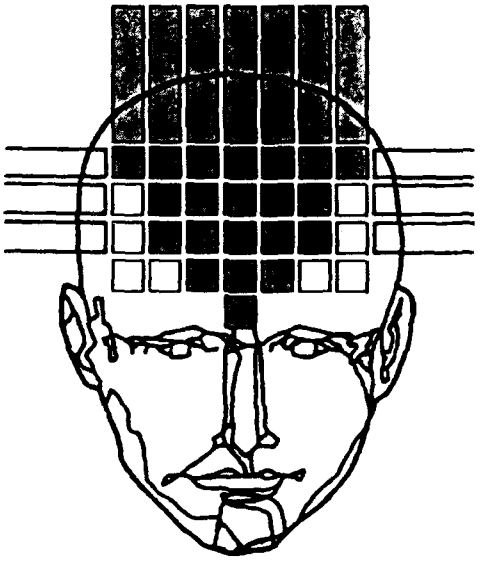
Navy Personnel Research and Development Center
San Diego, California 92152-7250 AP-93-10 August 1993

(4)

Conference
PROCEEDINGS

Applications of Artificial Neural Networks and Related Technologies to Manpower, Personnel, and Training



REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATE COVERED
	August 1993	Final--AP-93-10

4. TITLE AND SUBTITLE
Conference on Applications of Artificial Neural Networks and Related Technologies to Manpower, Personnel, and Training: Proceedings

5. FUNDING NUMBERS
PE 060223N, RM33M20/RM33T23

6. AUTHOR(S)
Jules Borack (Editor)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Navy Personnel Research and Development Center
San Diego, California 92152-7250

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NPRDC-AP-93-10

11. SUPPLEMENTARY NOTES
Functional Area: Personnel
Product Line: Personnel Classification
Effort: Classification Techniques

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT *(Maximum 200 words)*

This conference proceedings documents papers presented at the Artificial Neural Networks and Related Technologies to Manpower, Personnel, and Training Conference held at the Navy Personnel Research and Development Center on 2-3 February 1993. Conference participants represented the academic and military research communities, both governmental and private. Conference presentations discussed the relationship of neural networks to statistics, personnel, brain activity, expert systems, classification and learning, and tactical issues. A conference seminar discussed "Learning in Artificial Neural Networks: A Statistical Perspective."

14. SUBJECT TERMS
artificial neural networks, manpower, personnel, training, expert systems, classification, learning

15. NUMBER OF PAGES
181

16. PRICE CODE

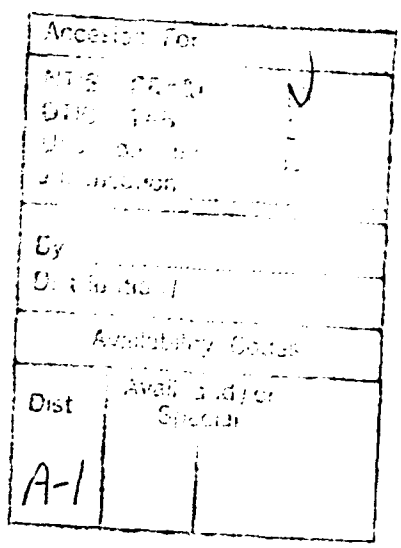
17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNLIMITED

Foreword

This conference on applications of Artificial Neural Networks and Related Technologies to Manpower, Personnel, and Training was held at the Navy Personnel Research and Development Center on 2-3 February 1993. The meetings were supported through the 6.2 Technology Development Block Program (PE 060223N). Conference participants represented the academic and military research communities, both governmental and private. The papers reflect the opinions of their authors only and are not to be construed as the official policy of any institution, government, or branch of the armed services.

JOHN D. McAFEE
Captain, U.S. Navy
Commanding Officer

RICHARD C. SORENSON
Technical Director (Acting)



Contents

Neural Networks, Expert Systems, Classification and Learning (Paper Session)

Neural Networks, Tactical and Related Issues (Paper Seminar)

The Future of Neural Networks (and Related Technologies) and their Application to Manpower, Personnel, and Training—Opportunities and Challenges (Discussion Groups)

Conference Information

Author Index

Welcome

Richard C. Sorenson
Technical Director, Navy Personnel Research and Development Center

I would like to welcome everyone to the Navy Personnel Research and Development Center. We have over thirty participants from the Air Force, Army and Navy, and another dozen or so of our Center researchers who maintain an interest in artificial neural networks and affiliated research areas. Both military and academic research communities are represented, covering a broad spectrum of R&D responsibilities from front-line researchers through R&D managers. Many of you here today have been acknowledged for your expertise in artificial neural networks and expert systems.

The first model of neural networks dates back fifty years to 1943; in this sense, conceptual models actually preceded technology. At that time, Warren McCulloch and Walter Pitts proposed a neural processing model drawn from information theory, which had recently been advanced by Alan Turing and Claude Shannon. These early pioneers coined the term, 'cybernetics' to define their new field of interest.

The progress made in the past fifty years is a good example of the interplay involving technology, knowledge (the cumulative advancement of technologies in several fields which expanded our understanding of biological mechanisms as well as our ability to examine processes within living organisms), and inquiry (bridging, integrating, and synthesizing). Initially, technology (in this case vacuum tube computers) lagged human concepts of neural models. Without sufficient technology, the predominant focus in cybernetics was research on artificial intelligence (expert systems), as defined by Minsky, Newell, and Simon. As computer capabilities expanded, including parallel processing, the technology became more aligned with researchers' thinking. By the early 1980s--just ten years ago--the resurgence of interest had a strong focus on neural networks in living systems and their emulation with high capacity computers (technology = Cray parallel processing computers).

There has been a substantial R&D investment in artificial neural networks and related technologies by the Defense Advanced Research Projects Agency (DARPA) and the Office of Naval Research (ONR) over the past five years. Applications to manpower, personnel, and training are specialized. This conference provides us with the opportunity to exchange views on the field and examine different applications ranging from discovery systems (Scheines; Carnegie Mellon) through neuroscience (Greg Lewis et al; NPRDC). We'd also like this to be an opportunity to do some brainstorming on current issues and future directions.

In closing, I'm sure we all stand to gain from the papers being presented. We're also hoping that during the intermissions and informal lunches, you'll all have opportunities to share experiences in your fields with one another.

LEARNING IN ARTIFICIAL NEURAL NETWORKS:

A STATISTICAL PERSPECTIVE *

by

Halbert White

August 1989

* The author is indebted to Mark Salmon for helpful comments and references.
This work was supported by National Science Foundation Grant SES-8806990.

ABSTRACT

The premise of this article is that learning procedures used to train artificial neural networks are inherently statistical techniques. It follows that statistical theory can provide considerable insight into the properties, advantages and disadvantages of different network learning methods. We review concepts and analytical results from the literatures of mathematical statistics, econometrics, systems identification and optimization theory relevant to the analysis of learning in artificial neural networks. Because of the considerable variety of available learning procedures and necessary limitations of space, we cannot provide a comprehensive treatment. Our focus is primarily on learning procedures for feedforward networks. However, many of the concepts and issues arising in this framework are also quite broadly relevant to other network learning paradigms. In addition to providing useful insights, the material reviewed here suggests some potentially useful new training methods for artificial neural networks.

1. INTRODUCTION

Readers of this journal* are by and large well aware of the widespread and often dramatic successes recently achieved through the application of connectionist modeling and learning techniques to an impressive variety of pattern recognition, classification, control and forecasting problems. In many of these cases, success has been achieved by the now rather simple expedient of appropriately training a hidden layer feedforward network using some variant of the method of back-propagation (Werbos [1974], Parker [1982], Rumelhart, et al. [1986]). These successes have stimulated an entire industry devoted to devising ever new and better variants of back-propagation. Typically, papers representative of this industry contain some clever heuristics and some more or less limited experiments demonstrating the advantages of the new and improved methods. These successes have also encouraged consideration of some important and difficult questions, such as, "Under what conditions will a given network generalize well?"; "What is meant by generalization?"; "How can one determine an appropriate level of complexity for a given network?"; "How can one tell when to stop training if the targets are affected by unmeasurable noise?"

The premise of this paper is that learning procedures used to train artificial neural networks are inherently statistical techniques. It follows that statistical theory can provide considerable insight into the properties, advantages and disadvantages of different network learning methods. The literature of statistics and the related literatures of systems identification and econometrics can suggest improvements to current approaches to network learning, as well as useful new approaches. Furthermore, these fields suggest additional important questions that should be asked in studying network

* Reprinted from *Neural Computation*, Volume 1, Number 4; Halbert White, "Learning in Artificial Neural Networks: A Statistical Perspective," by permission of the MIT Press, Cambridge, Massachusetts, Copyright 1989.

learning, but which have not yet been clearly formulated or widely appreciated in the connectionist literature. Examples of such questions are "Under what conditions do the weights generated by a given learning method converge as the size of the training set grows and to what do they converge?"; "What is the rate of convergence and how is this affected by the choice of the learning rate?"; "Can the limiting behavior of the learned weights be described by some known stochastic process?"; "Is a given hidden unit or input unit contributing to successful network performance or is it irrelevant?"; "Does a given learning procedure extract all the available statistical information contained in a given body of data, or is it statistically inefficient?" Answers to these questions are available from the theory of mathematical statistics; these answers are also relevant to the questions raised earlier.

The purpose of this article is to review concepts and analytical results from the literatures of mathematical statistics, systems identification and econometrics relevant to the analysis of learning in artificial neural networks, with particular attention paid to material bearing on the answers to the questions just raised. Because of the considerable variety of available learning procedures and necessary limitations of space, it will not be possible to provide a comprehensive treatment. Our focus here will be primarily on learning procedures for feedforward networks. However, many of the concepts and issues arising in this framework are also quite broadly relevant to other network learning paradigms. We comment on some of these as we proceed.

The plan of this paper is as follows. In Section 2, we show why it is that mathematical statistics has something to say about network learning. Section 3 discusses relevant concepts of probability fundamental to the analysis of network learning. In Section 4, we consider some alternative approaches to network learning and describe the

statistical properties of these methods. Section 5 provides a review of some recently obtained results establishing that multilayer feedforward networks are capable of learning an arbitrary mapping; these results apply recent developments in the nonparametric statistics literature. Section 6 contains a brief summary and some concluding remarks.

2. THE RELEVANCE OF STATISTICS

Suppose we are interested in learning about the relationship between two variables X and Y, numerical representations of some underlying phenomenon of interest. For example, X could be measurements on geological attributes of a site, and Y could be a variable assuming the value one if oil is present and zero otherwise. Alternatively, X could be measurements of various economic variables at a particular point in time and Y could be the closing value for the Dow Jones index on the next day. As another example, X could be the treatment level for an experimental drug in a controlled experiment using laboratory animals, and Y could be the percentage of the group treated at that level that benefit from the drug.

Often, a theory exists or can be constructed that describes a hypothesized relation between X and Y, but the ultimate success or failure of any theory must be determined by an examination of how well its predictions accord with repeated measurements on the phenomenon of interest. In other cases, no satisfactory theory exists or can be constructed because of the complexity of the phenomenon and the difficulties of controlling for difficult to measure influences that are correlated with measurable influences. Nevertheless, repeated measurements can be obtained on a subset of relevant variables (i.e. X and Y).

In either case, the possibility of making repeated measurements allows us to build up a form of empirical knowledge about the phenomenon of interest. A neural network is one form in which this empirical knowledge can be encoded.

The relevance of statistical analysis arises as soon as repeated measurements are made. Suppose we have n measurements or "observations" on X, denoted $x_1, ..., x_n$ and n corresponding observations on Y, denoted $y_1, ..., y_n$. Both X and Y may be vector quantities, and therefore so will be x_t and y_t, $t = 1, ..., n$. We suppose that X is of dimension $r \times 1$ and Y is of dimension $p \times 1$ for integers r and p. For notational convenience, we shall write $Z = (X', Y')'$ and $z_t = (x_t', y_t')'$, where a prime denotes vector (or matrix) transposition. Thus, we have n observations, denoted $z^n = (z_1, ..., z_n)$. We refer to z^n as a "sample", or in connectionist jargon, a "training set." It is convenient to suppose that the measurement process could be continued indefinitely, in which case we would obtain a sequence of observations $\{z_t\} = (z_t, t = 1, 2, ...)$.

By definition, a statistic is any function of the sequence $\{z_t\}$. A familiar example of a statistic is the sample average of observations on Y, $n^{-1} \sum_{t=1}^{n} y_t$. A less familiar example of a statistic, but one which provides a complete representation of our empirical knowledge is the sample itself, z^n (a matrix-valued statistic). Because the entire sample is an unwieldly way of representing our empirical knowledge, we can attempt to boil it down or summarize it in some convenient way, which is why such things as averages and correlations ("summary statistics") are useful. However, a potentially much more powerful way of boiling down our empirical knowledge is to convert it into the weights of a suitable neural network. Because this conversion can only be accomplished as some function of the sequence $\{z_t\}$, the resulting network weights are a (vector-valued)

statistic. Thus, the process of network learning on a given training set is in fact the process of computing a particular statistic.

It follows that the analytical tools that describe the behavior of statistics generally can be used to describe the behavior of statistics specifically obtained by some neural network learning procedure. These behavioral properties all have a fundamental bearing on the answers to the questions posed in Section 1.

These considerations are quite general. They apply regardless of whether we consider artificial neural networks and learning algorithms such as back-propagation or Kohonen self-organization, or whether we consider biological neural networks and whatever actual learning mechanisms occur there. Because the latter are largely unknown, our subsequent focus will be on learning in artificial neural systems. However, the concepts relevant for examining artificial systems are also relevant for the study of natural systems.

3. PROBABILISTIC BACKGROUND

3.a Measurements, Probability Laws and Conditional Probability

Consideration of the method by which our measurements are obtained is fundamental to the analysis of any resulting statistics. It is helpful to distinguish initially between cases in which we have complete control over the values x_t and those cases in which we do not, and between cases in which y_t is determined completely by the values x_t and those cases in which other influences affect the measurement y_t.

Situations in which we have complete control over the values taken by x_t occur in the laboratory when it is possible to set experimental conditions with absolutely perfect

precision or occur in computer experiments. Situations in which control is not complete are common; these occur when nature has a hand in generating measurements x_t. Nature's role may be complete, as in the social sciences or meteorology, or it may be partial, as when our measurements are gathered by stratified sampling of some population or in an experiment in which, although x_t can be measured with absolute precision, its precise value is determined to some extent by chance.

In either of these situations, it is possible and quite useful to define an "environmental" probability law μ (or simply an "environment") that provides a complete description of the manner in which the measurements are generated. When nature's role in determining x_t is complete, we can regard x_t as a realization of the random variable X_t having probability law μ. When the experimenter's control is complete, we can regard μ as describing the relative frequencies with which different values for x_t are set. When the researcher has partial experimental control (but still perfect measurement capability) we can regard μ as embodying the combined influences of both nature and the experimenter, again determining the relative frequencies with which different values for x_t are observed. Formally, μ assigns to every relevant subset of $\mathbb{R}^r$ a number between zero and one representing the relative frequency with which x_t is observed to belong to that subset.

Now consider the determination of y_t. Cases in which y_t is determined completely by x_t occur in computer experiments or in physical systems in which every single influence in the determination of y_t can be measured with perfect precision and there is no inherent uncertainty attaching to y_t. For these cases, we can express an exact functional relationship between x_t and y_t as

$$y_t = g(x_t)$$

for some mapping $g : \mathbb{R}^r \rightarrow \mathbb{R}^p$. The function g embodies everything there is to know about the relationship between y_t and x_t. It is the mapping g that is the natural object of interest in this case.

In any situation in which it is not possible to obtain absolutely precise measurements on every single influence affecting the measurement y_t or in which y_t is subject to inherent uncertainty, it is no longer possible to express an exact functional relationship between x_t and y_t. Instead it is possible to express a *probabilistic* relationship between x_t and y_t. For this, it is appropriate to view x_t and y_t as a realization of the jointly distributed random variables X_t and Y_t. For notational convenience, we write $Z_t = (X_t', Y_t')'$. Hence Z_t is a random vector with $r + p$ components. Just as with X_t, we can define a joint probability law ν that describes the relative frequency of occurrence of vectors Z_t. The law ν embodies the environment μ as well as the probabilistic relationship between X_t and Y_t. Because we shall assume that X_t and Y_t have the same joint probability law as X and Y, we drop the subscript t whenever convenient.

The probabilistic relationship between X and Y is completely summarized by the conditional probability law of Y given X, which we denote as $\gamma(\cdot \mid x)$, i.e. $\gamma(A \mid x) = P[Y \in A \mid X = x]$, for any set A in $\mathbb{R}^p$. The notation $P[Y \in A \mid X = x]$ is read as "the probability (P) that Y belongs to ($\in$) the set A given that ($\mid$) the random variable X takes on (=) the value x."

In the case where Y is completely determined by X, e.g. as $Y = g(X)$, we have $P[Y = g(x) \mid X = x] = \gamma([g(x)] \mid x) = 1$ for all x. (We denote the set consisting of the single element $g(x)$ as $[g(x)]$.) Otherwise, there is generally no function g such that $\gamma([g(x)] \mid x) = 1$ for all x.

Because a proper understanding of the notions just introduced is important for following the discussion ahead, we shall briefly summarize before proceeding. The foregoing discussion establishes that a single framework applies to all the different situations initially distinguished at the beginning of this section. This framework is that the joint behavior of X and Y is described by a joint probability law ν. (True randomness is allowed, but not required.) This joint behavior can be decomposed into a probability law μ (the "environment", that describes the behavior (relative frequency of occurrence) of X, and a conditional probability law γ that describes the behavior (relative frequency of occurrence) of Y given X. In this "probabilistic" context, it is the knowledge of γ that is the natural object of interest, because this function embodies all there is to know about the relationship of interest, that between X and Y.

The case in which there is an exact functional relationship g is a special case; in this case, knowledge of γ and knowledge of g are equivalent. The relevance of the probabilistic context is that it applies to a much wider class of phenomena, as the discussion at the beginning of this section should suggest. Accordingly, from now on we take γ to be the fundamental object of interest in our study of the relationship between X and Y.

Certain aspects of the conditional probability law γ play an important role in interpreting what it is that is learned by artificial neural networks using standard techniques. Primary among these is the conditional expectation of Y given X, denoted $E(Y \mid X)$. This conditional expectation gives the value of Y that will be realized "on average", given a particular realization for X. Whenever $E(Y \mid X)$ exists, it can be represented solely as a function of X, i.e. $g(X) = E(Y \mid X)$ for some mapping $g : \mathbb{R}^r \rightarrow \mathbb{R}^p$. The expected value for Y given that we observe a realization x of X is then $g(x)$. Of

course, this value will be correct only "on average." The actual realization of Y will almost always differ from $g(x)$. We can define a random "expectational error" $\varepsilon \equiv Y - E(Y \mid X)$. Because $g(X) = E(Y \mid X)$ we can also write

$$Y = g(X) + \varepsilon.$$

By definition of ε and by the properties of conditional expectation, it follows that $E(\varepsilon \mid X) = 0$. That is, the average expectational error given any realization of X is zero. This contains the previous case of an exact relationship as the special case in which $\varepsilon = 0$ for all realizations of X. With a probabilistic relationship, ε is non-zero with positive probability.

An important special case occurs when Y can take on only the values 0 and 1, as is appropriate for any two-way classification problem. For this case, the conditional expectation function g also provides the conditional probability that $Y = 1$ given any realization of X, i.e. $\gamma([1] \mid x) = g(x)$. Because the conditional probability embodies all information available about the relationship between X and Y, a knowledge of g provides complete knowledge about the phenomenon of interest for the classification problem just as it does in the case of exact determination of Y by X.

This discussion highlights some of the reasons for the theoretical importance of the conditional expectation function. The reason for its important role in network learning will become apparent when we subsequently examine specific learning methods.

3.b Objectives of Learning

Although the conditional probability law γ is a natural object of interest in the abstract, learning in neural networks, whether natural or artificial, is not necessarily

directed in an explicit manner at the discovery of γ. Instead, a common goal of network learning is that the network perform acceptably well (or even optimally) at some specific task in a specific environment. However, because of the fundamental role played by γ, such performance-based objectives for network learning typically are equivalent or closely related to methods explicitly directed at the discovery of particular specific aspects of γ. We examine this linkage in this section.

When the relationship between X and Y is of interest, it is often because X is to be used to predict or explain Y. In such cases, network performance can be measured using a performance function $\pi : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$. Given a target value y and network output o, the performance function gives a numerical (real-valued) measure of how well the network performs as $\pi(y, o)$. It is convenient to normalize π so that the bigger $\pi(y, o)$ is, the worse is the performance.

A frequently encountered performance measure for artificial neural networks is squared error,

$$\pi(y, o) = | y - o |^2 / 2.$$

Many other choices are possible: with $p = 1$ (y and o now scalars) we could also take $\pi(y, o) = | y - o |$, $\pi(y, o) = | y - o |^q / q$, or $\pi(y, o) = -[y \log o + (1 - y) \log (1 - o)]$ (for $0 < o < 1$). Note that in each of these cases $\pi(y, o) \geq 0$ and $\pi(y, o)$ is minimized if and only if $y = o$. Such behavior for π is often convenient, but is not an absolute requirement; we might wish to make π measure the profit in dollars made by action o of the network when the environment produces a realization y.

Network output can generally be expressed in terms of an output function mapping inputs and network weights into network output. Formally, $f : \mathbb{R}^r \times W \to \mathbb{R}^p$, where W is

a weight space appropriate to the network architecture embodied in f. We take W to be a subset of $\mathbb{R}^s$, where s is some integer. The precise form of f is not of particular importance. Given weights w and inputs x, output is given as $o = f(x, w)$. Given targets y, network performance is then $\pi(y, f(x, w))$.

For any combination of y and x, and for any choice of weights w, we can now measure network performance (as $\pi[y, f(x, w)]$); however, it is generally required that a network perform well in a range of situations, i.e. for a range of different values for y and x. One way of making this requirement precise is to require the network to perform well "on average." Average performance is given mathematically by the (unconditional) expectation of the random quantity $\pi(Y, f(X,w))$, expressed formally as

$$\lambda(w) \equiv \int \pi\,(y, f(x, w))\, \nu\,(dy, dx)$$

$$\equiv E[\pi(Y, f(X, w))], \quad w \in W$$

We call λ the "expected performance function." Note that it depends only on the weights w, and not on particular realizations y and x. These have been "averaged out." This averaging is explicit in the integral representation defining λ. The integral is a Lebesgue integral taken over $\mathbb{R}^{p+r}$. The Lebesgue measure ν permits integrating either continuous or discrete measures (or a mixture of the two) over $\mathbb{R}^{p+r}$. The second expression reflects the fact that averaging $\pi(y, f(x, w))$ over the joint distribution of Y and X, i.e. ν, gives the mathematical expectation ($E(\cdot)$) of the random performance $\pi(Y, f(X, w))$.

Because we are concerned with artificial networks, we have the potential capability of selecting weights w that deliver the best possible average performance. In the context of artificial networks, then, it is sensible to specify that the goal of network learning is to find a solution to the problem

$$\min_{w \in W} \lambda(w).$$

We denote the solution to this problem w^*, the "optimal weights."

The requirement that λ represent average performance is imposed above for concreteness, not out of necessity. We shall continue to use this interpretation, but it should be realized that λ may more generally represent any criterion (e.g. median performance) relevant in a given context.

To illustrate our earlier remark that choosing a performance measure π is intimately related to which aspect of the probabilistic relationship between X and Y is implicitly of concern, consider the case in which $\pi(y, o) = (y - o)^2$. Then

$$\lambda(w) = E([Y - f(X, w)]^2).$$

Taking $g(X) = E(Y \mid X)$, we have

$$\lambda(w) = E([Y - g(X) + g(X) - f(X, w)]^2)$$

$$= E([Y - g(X)]^2) + 2\,E([g(X) - f(X, w)]\,[Y - g(X)]) + E([g(X) - f(X, w)]^2)$$

$$= E([Y - g(X)]^2) + E([g(X) - f(X, w)]^2).$$

The final equality holds because $E([g(X) - f(X, w)]\,[Y - g(X)]) = E([g(X) - f(X, w)]\varepsilon) = E[E([g(X) - f(X, w)]\varepsilon \mid X)] = E[(g(X) - f(X, w))\,E(\varepsilon \mid X)] = 0$ by the law of iterated expectations and the properties of ε noted earlier. It follows that w^* not only minimizes $\lambda(w)$, but also minimizes

$$E([g(X) - f(X, w)]^2) = \int [g(x) - f(x, w)]^2 \mu(dx).$$

In other words, w^* is the weight vector having the property that $f(\cdot, w^*)$ is a mean-squared error minimizing approximation to the conditional expectation function g. It is

this aspect of the probabilistic relationship that becomes the focus of interest under the squared error performance measure.

Note that the *environment measure* μ *plays a crucial role here in the determination* of the optimal approximating weights w^*. These weights give small errors (on average) for values of X that are very likely to occur at the cost of larger errors (on average) for values of X that are unlikely to occur. It follows that the weights w^* will not give optimal performance in an operating environment $\tilde{\mu} \neq \mu$. This crucial role holds generally, not just for the case of squared error.

A similarly crucial role in the determination of w^* is played by the performance function π. Weights w^* optimal under the choice π need not be optimal for some other performance measure $\tilde{\pi} \neq \pi$. If performance in the operating environment is to be evaluated using $\tilde{\pi} \neq \pi$ (e.g. maximum absolute error instead of mean squared error), then weights w^* will not give optimal operating performance. Consequently, it is of great importance that π and μ be selected so as to reflect accurately the conditions under which operating performance is to be evaluated. Suboptimal network performance will generally result otherwise.

By taking the weights w^* to be the object of network learning, we automatically provide a solution to the question of what is meant by "generalization." Weights w^* generalize optimally *by construction* in the sense that given a random drawing from the probability law ν governing X and Y, network output $f(X, w^*)$ has the best average performance, $\lambda(w^*)$. As long as ν governs the observed realization, a given random drawing need not have been "seen" by the network before. On the other hand, if the realization is drawn from an environment different than that from which the optimal w^* is

obtained, then the network will not generalize as well as it could have in this precise sense, even if it has "seen" the particular realization during training.

Further interpretation of the average performance function is possible. In particular, put

$$h(y; x, w) = k_o(x, w)^{-1} \exp[-\pi(y, f(x, w))],$$

where we assume that $k_o(x, w) = \int \exp[-\pi(y, f(x, w))]\gamma(dy \mid x)$ is finite. It follows that for each x and w, $h(\cdot; x, w)$ is a conditional probability density function on $\mathbb{R}^p$. This can be viewed as an approximation to the true conditional probability density $d\gamma(\cdot \mid x)$ of the conditional probability law $\gamma(\cdot \mid x)$. Taking logarithms gives

$$\log h(y; x, w) = -\log k_o(x, w) - \pi(y, f(x, w)).$$

Thus

$$\begin{aligned}
\lambda(w) &= \int \pi(y, f(x, w))\, \nu(dy, dx) \\
&= \int [\int -\log h(y; x, w)\gamma(dy \mid x)]\mu(dx) - \int \log k_o(x, w)\mu(dx) \\
&= \int [\int \log [d\gamma(y \mid x) / h(y; x, w)]\gamma(dy \mid x)]\mu(dx) \\
&\quad - \int \log k_o(x, w)\mu(dx) - \int \log d\gamma(y \mid x)\nu(dy, dx).
\end{aligned}$$

The term in brackets in the first integral, which we define as

$$I(d\gamma : h; x, w) \equiv \int \log [d\gamma(y \mid x) / h(y; x, w)]\gamma(dy \mid x)$$

is the *Kullback-Leibler Information of* $h(\cdot; x, w)$ *relative to* $d\gamma(\cdot \mid x)$ (Kullback and Leibler [1951]). This is a fundamental information theoretic measure of how accurate the conditional density $h(\cdot; x, w)$ is as an approximation to the true conditional density $d\gamma(\cdot \mid x)$ (see e.g. Renyi [1961]). Heuristically, $I(d\gamma : h; x, w)$ measures the information

theoretic surprise we experience when for given x and w we believe the conditional density of Y given X is $h(\cdot; x, w)$ and we are then informed that the conditional density is in fact $d\gamma(\cdot \mid x)$. A fundamental theorem is that $I\!I(d\gamma : h; x, w) \geq 0$ for all x and w and that $I\!I(d\gamma : h; x, w) = 0$ if and only if $d\gamma(y \mid x) = h(y; x, w)$ for almost all y (under $\gamma(\cdot \mid x)$). In other words, this information measure is never negative, and is zero when (and only when) $h(\cdot; x, w)$ is in fact the true conditional density.

Substituting, we have

$$\lambda(w) = E(I\!I(d\gamma : h; X, w)) + k_1(w),$$

where $k_1(w) \equiv -\int \log k_o(x, w)\mu(dx) - \int \log d\gamma(y \mid x)\nu(dy, dx)$. In the important and common case where $k_1(w)$ is constant as a function of w (i.e. whenever $k_o(x, w)$ does not depend on w), the average performance function can be interpreted as differing by a constant from the expected Kullback-Leibler Information of the conditional density $h(\cdot : x, w) \equiv k_o(x, w)^{-1} \exp[-\pi(\cdot, f(x, w))]$ relative to the true conditional density $d\gamma(\cdot \mid x)$.

It follows that the optimal weights w^* have a fundamental information theoretic interpretation, in that they minimize expected Kullback-Leibler Information given the chosen architecture (embodied by f) and performance measure π. Further, when $E(I\!I(d\gamma : h; X, w^o)) = 0$ for some w^o in W it follows that $d\gamma(y \mid x) = h(y; x, w^o)$ *a.s.*-ν and $w^* = w^o$. Thus w^* indeed provides complete information on the probabilistic relation between X and Y if this is possible given f and π. Further general discussion of the meaning of Kullback-Leibler Information in a related context is given in White [1989a, ch. 2-5]. Viewing learning as related to Kullback-Leibler Information in this way implies that learning is a *quasi-maximum likelihood* statistical estimation procedure. White [1989a] contains an extensive discussion of this subject.

It is important to emphasize that none of the discussion in this section depends very much on the particular neural network architecture under consideration, or even on the use of a neural network model at all. The foregoing considerations pertain equally well to any statistical modeling procedure in which the target Y is approximated by $f(X, w)$. This is the common situation in all of parametric statistics. The role of neural network modeling is to provide a specific form for the function f. The advantages of neural network modeling have to do with the virtues associated with such specific forms. We shall return to this point again later.

In many network paradigms, there may be no particular target, or the target and input may be the same. Nevertheless, it is often still possible to define a learning objective function as

$$\lambda(w) = \int l(z, w)\nu(dz)$$

where $l : \mathbb{R}^{p+r} \times W \to \mathbb{R}$ is a given loss function measuring network performance given weights w (the state of the network) and observables z (the state of the world). The interpretation of learning now is that the goal of the network is to adjust its state (w) in such a way as to minimize the expected loss suffered over the different possible states of the world (z). In the special case in which targets and inputs are distinguished and π is given as above, we have $l(z, w) = \pi(y, f(x,w))$. We shall make use of the general formulation in terms of loss functions in what follows, although our examples will typically assume distinct targets and inputs.

4. STATISTICAL PROPERTIES OF LEARNING METHODS

We saw in the previous section that the goal of network learning can be viewed as finding the solution w^* to the optimization problem

$$\min_{w \in W} \lambda(w) = \int l(z, w) \nu(dz). \tag{4.1}$$

If the joint probability law ν were known, w^* could be solved for directly. It is our ignorance of ν that makes learning necessary. It is the nature of the response to this ignorance that leads to specific learning algorithms. The details of these algorithms interact with the probability law (call it P) governing the entire sequence $\{Z_t\}$ to determine the properties of given learning algorithms. It is the role of statistical analysis to describe these properties. In this section, we describe several possible responses and the statistical properties of the resulting learning algorithms.

Because we are concerned with artificial neural networks, we are not limited to learning methods that have biological or cognitive plausibility. Thus, we are free to consider "artificial" learning methods. To the extent that biological or cognitive processes or constraints suggest useful approaches to learning (i.e. solving the problem (4.1)), we are free to adopt them. To the extent that such processes or constraints get in the way of using an artificial network to encode empirical knowledge, we are free to dispense with them. As we shall see, basing our approach to learning on the principles of analytical and computational expediency nevertheless leads us to an appreciation of the usefulness of such methods as back-propagation, simulated annealing and the genetic algorithm.

4.a Learning by Optimizing Performance Over the Sample

Despite our fundamental ignorance of ν, our ability to make repeated measurements on $Z = (X', Y')'$ permits us to obtain empirical knowledge about ν. Given a sample z^n (recall $z^n = (z_1, ..., z_n)$), a direct sample analog of ν, denoted ν_n, can be calculated as

$$\nu_n(C) \equiv (\#\text{ of times } z_i \text{ belongs } C) / n,$$

where C is any subset of $\mathbb{R}^{p+r}$. When n is large, the law of large numbers ensures that this will be a good approximation to $\nu(C)$ for any set C. Using this approximation to ν, it is possible to compute an approximation to λ as

$$\lambda_n(w) \equiv \int l(z, w)\nu_n(dz)$$

$$= n^{-1} \sum_{i=1}^{n} l(z_i, w), \quad w \in W.$$

This is easily recognized as average performance of the network over the sample (training set). Because this number is readily computed, we can attempt to solve the problem

$$\min_{w \in W} \lambda_n(w).$$

We denote the solution to this problem as w_n. We make no attempt to justify the attempt to solve this problem on biological or cognitive grounds, for the reasons given above. Consideration of this problem is helpful, however, because it delivers direct and deep insights and suggests useful practical learning methods. With this approach, the study of network learning now reduces to the study of the relationship between w_n and w^*.

Before turning to the challenges that arise in attempting to solve this apparently feasible problem, we must first discuss a number of relevant issues. First, we must

recognize that we can in general say nothing about the precise relation between w_n and w^*. The problem is that w_n is a realization of random variable. The best that we can do is to make probability statements about the random variable giving rise to w_n. To obtain this random variable, we make use of the random counterpart of ν_n,

$$\hat{\nu}_n(C) \equiv (\#\text{ of times } Z_t \text{ belongs } C) / n, \quad C \subset \mathbb{R}^{p+r}$$

to define

$$\hat{\lambda}_n(w) \equiv \int l(z, w)\hat{\nu}_n(dz)$$

$$= n^{-1} \sum_{t=1}^{n} l(Z_t, w), \quad w \in W.$$

We then define $\hat{w}_n$ as the random variable that solves the problem

$$\min_{w \in W} \hat{\lambda}_n(w) = n^{-1} \sum_{t=1}^{n} l(Z_t, w). \tag{4.2}$$

In the special case where $\pi(y, o) = (y - o)^2 / 2$ we get $l(z, w) = (y - f(x, w))^2 / 2$ and

$$\min_{w \in W} \hat{\lambda}_n(w) = n^{-1} \sum_{t=1}^{n} (Y_t - f(X_t, w))^2 / 2. \tag{4.3}$$

This is precisely the problem of nonlinear least squares regression, which has been extensively analyzed in the econometrics, statistics and systems identification literatures. We give some relevant references below.

Thus, the solution w_n defined earlier is simply a realization of the random variable $\hat{w}_n$. Consequently, we focus attention on the relationship between the random variable $\hat{w}_n$ and the optimal weights w^*.

4.a(i) Large Sample Behavior of $\hat{w}_n$

As with any random variable, the behavior of $\hat{w}_n$ is completely described by its probability law. In general, this probability law is prohibitively difficult to obtain for a training set of given size n. However, approximations to this probability law for large n can be obtained by making use of standard statistical tools, including the law of large numbers and the central limit theorem.

These approximations reveal that the probability law of $\hat{w}_n$ collapses, i.e. becomes more and more concentrated, as n increases. The value around which this concentration occurs is therefore of fundamental importance. This increasing concentration property is referred to as the property of "consistency." We describe this in more detail below.

The collapse of the probability law of $\hat{w}_n$ can be shown to occur at a certain specific rate. It is possible to offset this collapse by a simple standardization. The approximate probability law of the standardized random variable is thus stabilized; this probability law is known as the "limiting distribution" or "asymptotic distribution" of $\hat{w}_n$. The central limit theorem is fairly generally applicable and ensures that the appropriate standardization of $\hat{w}_n$ has approximately a multivariate normal distribution. The approximation is better the larger is n. We describe this also in somewhat more detail below.

The fact that the limiting distribution of $\hat{w}_n$ is known has deep and far-reaching implications. In particular, this makes possible formal statistical inference regarding w^*. Because many questions of interest regarding the precise form of the optimal network architecture can be formulated as formal hypotheses regarding w^*, these questions can be resolved to the extent permitted by the available data by calculating some standard and

relatively straightforward statistics. To date, the profound significance of this fact has not been widely appreciated or exploited in the neural network literature. Below, we discuss some of the possible applications of these methods.

In the statistics literature, an examination of the consistency and limiting distribution properties of any proposed new statistic is standard. Such analyses reveal general useful properties and difficulties that are impossible to infer from or substantiate with Monte Carlo simulations. As the field of neural computation matures, rigorous analysis of the consistency and limiting distribution properties of any proposed new learning technique should become as standard as the Monte Carlo studies now prevalent. The discussion to follow will indicate some of the typical issues involved in such analyses.

4.a(ii) Notions of Stochastic Convergence

With this preview of where we are headed, let us return to the issue of consistency. There are three concepts that are directly relevant. The first is the standard concept of deterministic convergence. Let $\{a_n\} \equiv (a_1, a_2, ...)$ be a sequence of (non-random) real variables. We say that a_n *converges to* a, written $a_n \to a$ (as $n \to \infty$) if there exists a real number a such that for any (small) $\varepsilon > 0$, there exists an integer N_ε sufficiently large that $| a_n - a | < \varepsilon$ for all $n \geq N_\varepsilon$. We call a the "limit" of $\{a_n\}$.

Next, let $\{\hat{a}_n\}$ be a sequence of real-valued random variables. We say that $\hat{a}_n$ *converges to a almost surely* $-P$, written $\hat{a}_n \to a$ (as $n \to \infty$) *a.s.*$-P$ if $P[\hat{a}_n \to a] = 1$ for some real number a. That is, the probability of the set of realizations of $\hat{a}_n$ for which (deterministic) convergence to a occurs has probability 1. Heuristically, it is possible for a realization of $\{\hat{a}_n\}$ to fail to converge, but it is more likely that all the ink on this page

will quantum mechanically tunnel to the other side of the page sometime in the next five femtoseconds. This form of stochastic convergence is known as "strong consistency" or "convergence with probability one" (*w.p.* 1.). It is also written as $\hat{a}_n \overset{a.s.}{\to} a$.

A weaker form of stochastic convergence is convergence "in probability." Again, let $\{\hat{a}_n\}$ be a sequence of random variables. We say that $\hat{a}_n$ *converges to a in probability* $(-P)$, written $\hat{a}_n \to a \; prob-P$ if there exists a real number a such that for any (small) $\varepsilon > 0, P[\mid \hat{a}_n - a \mid < \varepsilon] \to 1$ as $n \to \infty$. Heuristically, the probability that $\hat{a}_n$ will be found within ε of a tends to one as n becomes arbitrarily large. This form of stochastic convergence is known as "weak consistency." It is implied by strong consistency. Convergence in probability is also written $\hat{a}_n \overset{P}{\to} a$.

Applying these concepts to $\hat{w}_n$, it would be satisfying to establish that $\hat{w}_n \to w^* \; a.s.-P$, i.e. that $\hat{w}_n$ is strongly consistent for w^*. This is in fact true under general conditions on l, W and $\{Z_t\}$ discussed in the econometrics literature by White [1981, 1982, 1984a, 1989] and Domowitz and White [1982].

It is useful to give a brief description of the underlying heuristics. The basic idea is that because $\hat{\lambda}_n(w)$ is an average of random variables (for each fixed w), the law of large numbers applies (under conditions placed on the probability law P governing $\{Z_t\}$, and on l -- see White [1984b, ch. 2]) to ensure that $\hat{\lambda}_n(w) \to \lambda(w) \; a.s.-P$. Because $\hat{w}_n$ minimizes $\hat{\lambda}_n$ and w^* minimizes λ and because $\hat{\lambda}_n$ and λ are close $a.s.-P$ for n large, then $\hat{w}_n$ should be close to w^*. This heuristic argument is not complete, but it can be made complete by ensuring that the convergence of $\hat{\lambda}_n$ to λ is uniform over W (i.e. $\sup_{w \in W} \mid \hat{\lambda}_n(w) - \lambda(w) \mid \to 0 \; a.s.-P$). For this, it helps to assume that W is a compact set.

These issues have been thoroughly studied in the econometrics literature under general conditions. It follows under very general conditions on P, l and W that any learning procedure capable of successfully solving the problem (4.2) delivers learned weights $\hat{w}_n$ that are arbitrarily close to the optimal weights w^* for all n sufficiently large, with probability one. This provides a definitive answer to the question of what it is that networks learn when (4.2) is solved.

The limiting distribution of $\hat{w}_n$ is studied in the same references. The appropriate formal concept is that of convergence in distribution. Let $\{\hat{a}_n\}$ be a sequence of random variables having distribution functions $\{F_n\}$ (recall $F_n(a) \equiv P[\hat{a}_n \leq a]$). We say that $\hat{a}_n$ *converges to F in distribution*, written $\hat{a}_n \xrightarrow{d} F$, if and only if $| F_n(a) - F(a) | \to 0$ for every continuity point a of F. This is a very weak convergence condition indeed. However, it permits approximately accurate probability statements to be made about $\hat{a}_n$ using the limiting distribution F in place of the exact distribution F_n. The ability to make such probability statements is quite useful.

Under fairly general conditions, the central limit theorem can be applied to establish that the limiting distribution of $\sqrt{n}(\hat{w}_n - w^*)$ is the multivariate normal distribution with mean vector zero and an $s \times s$ covariance matrix (say V^*) that can be given a precise analytic expression. We refer to V^* as the "asymptotic covariance matrix" of $\hat{w}_n$. The smaller is this covariance matrix (as measured by, say, $tr\, V^*$ or $\det V^*$) the more tightly the distribution of $\hat{w}_n$ is concentrated around w^*, with less consequent uncertainty about the value of w^*. It is therefore desirable that V^* be small, but there are fundamental limits on how small V^* can be. When two learning methods yield weights $\hat{w}_{1n}$ and $\hat{w}_{2n}$ respectively that are both consistent for w^*, one with asymptotic covariance matrix V_1^*,

and other with asymptotic covariance matrix V_2^*, the method yielding the smaller asymptotic covariance matrix is preferable, because that method makes relatively more "efficient" use of the same sample information. In certain cases, it can be shown that $V_1^* - V_2^*$ is a positive semi-definite matrix, in which case it is said that the second method is "asymptotically efficient" relative to the first method. Thus, study of the limiting distribution of alternative learning methods can yield insight into the relative desirability of different learning methods. As a specific example, White [1989b] proves that learning methods that solve (4.2) for squared error performance are asymptotically efficient relative to the method of back-propagation. In this sense the method of back-propagation is statistically inefficient. Kuan and White [1989] discuss a modification of back-propagation that has asymptotic efficiency equivalent to the solution of (4.2).

4.a(iii) Statistical Inference and Network Architecture

Of significant consequence is the fact that the limiting distribution of $\hat{w}_n$ can be used to test hypotheses about w^*. Two hypotheses of particular importance in artificial neural networks are the "irrelevant input hypothesis" and the "irrelevant hidden unit hypothesis." The irrelevant input hypothesis states that a given input or group of inputs is of no value (as measured by λ) in predicting or explaining the target. The alternative hypothesis is that the given input or some member of the given group of inputs is indeed of value in predicting or explaining the target. Similarly, the irrelevant hidden unit hypothesis states that a given hidden unit or group of hidden units is of no value in predicting or explaining the target. The alternative hypothesis is that the given hidden unit or some member of the given group of hidden units is indeed of value in predicting or explaining the target. Because these hypotheses can generally be expressed as the restriction that particular

elements of w^* are zero (those corresponding to the specified units) and because the learned weights $\hat{w}_n$ are close to w^* for large n, the learned weights can be used to provide empirical evidence in favor of or in refutation of the hypothesis under consideration.

Under the irrelevant input hypothesis, the corresponding learned weights $\hat{w}_n$ should be close to zero. The question of how far from zero is too far from zero to be consistent with the irrelevant input hypothesis can be answered approximately for large n by making use of the known limiting distribution of $\hat{w}_n$.

Specifically, the irrelevant input hypothesis can be expressed as $H_o : Sw^* = 0$, where S is a $q \times s$ selection matrix picking out the q elements of w^* hypothesized to be zero under the irrelevant input hypothesis. The fact that $\sqrt{n}\,(\hat{w}_n - w^*)$ has a limiting multivariate normal distribution with mean zero and covariance matrix V^* implies that $\sqrt{n}\,S(\hat{w}_n - w^*)$ has a limiting multivariate normal distribution with mean zero and covariance matrix SV^*S'. Because the irrelevant input hypothesis implies $Sw^* = 0$, it follows that $\sqrt{n}\,S\hat{w}_n$ has a limiting multivariate around distribution with mean zero and covariance matrix SV^*S' under the irrelevant input hypothesis H_o. From this it follows that under H_o the random scalar $n\hat{w}_n'S'(SV^*S')^{-1}S\hat{w}_n$ has a limiting chi-squared distribution with q degrees of freedom (χ^2_q).

A realization of this random variable cannot be computed, because although an analytical expression for V^* is available, a knowledge of the probability law P is required for its numerical evaluation. Fortunately, an estimator of V^* can be constructed that is weakly consistent, i.e. there exists $\hat{V}_n$ such that $\hat{V}_n \to V^*$ prob-P. Replacing V^* with its weakly consistent estimator $\hat{V}_n$ has no effect on the limiting distribution of the statistic just given. Thus

$$n\hat{w}_n' S'(S\hat{V}_n S')^{-1} S\hat{w}_n \xrightarrow{d} \chi_q^2$$

under the irrelevant input hypothesis H_o. The probability distribution of the irrelevant input test statistic $n\hat{w}_n' S'(S\hat{V}_n S')^{-1} S\hat{w}_n$ is therefore well approximated for large n by the χ_q^2 distribution when the irrelevant input hypothesis is true. Under the alternative hypothesis $H_a : Sw^* \neq 0$, the irrelevant input test statistic tends to infinity with probability one. It follows that the procedure of failing to reject H_o whenever $n\hat{w}_n' S'(S\hat{V}_n S')^{-1} S\hat{w}_n$ fails to exceed the $1-\alpha$ percentile of the χ_q^2 distribution (for some typically small value of α, say $\alpha = 0.5$ or $\alpha = 0.01$) leads to incorrect rejection of the irrelevant input hypothesis with (small) probability approximately equal or less than α. As n becomes large the probability of correctly rejecting the irrelevant input hypothesis with this procedure tends to one (the test is "consistent"). This procedure is an application of standard techniques of statistical inference. It allows us to determine whether specific input(s) are irrelevant, to the extent permitted by the sample evidence by controlling the probability of incorrectly rejecting H_o. This approach has obvious applications in investigating the appropriateness of given network architectures.

The irrelevant hidden unit hypothesis is of exactly the same form, i.e. $H_o : Sw^* = 0$, except that now the $q \times s$ selection matrix S picks out weights associated with q hidden units hypothesized to be irrelevant. As before, the alternative is $H_a : Sw^* \neq 0$. Similar reasoning can be used to develop an irrelevant hidden unit test statistic. However there are some rather interesting difficulties in the development of the limiting distribution of $\hat{w}_n$ under H_o. Problems arise because when H_o is true, the optimal weights from input units to the irrelevant hidden unit(s) are not locally unique -- they have no effect on network output. This problem is known in the statistics literature as that in which

"nuisance parameters are identified only under the alternative hypothesis." Limiting distributions for such cases have been studied by Davies [1977, 1987]; the analysis is complicated. The resulting distributions are generally not χ^2. However, certain techniques can be adopted to avoid these difficulties, yielding a χ^2_q statistic for testing the irrelevant hidden unit hypothesis. One such test is described by White [1989c], and its properties are investigated by Lee, White and Granger [1989].

Statistical inference plays a fundamental role in modern scientific research. The techniques just described permit application of the methods of statistical inference to questions regarding the precise form of optimal artificial neural network architectures.

4.a(iv) Methods for Optimizing Performance Over the Sample

Now that we have at least superficially explored the consistency and limiting distribution properties of solutions to (4.2) and the implications of these properties, we may consider how (4.2) might be solved in practical situations. In general, we seek a global solution to what is typically a highly nonlinear optimization problem. Such problems are the general concern of an entire sub-area of mathematics, optimization theory. Rinnooy Kan, Boender and Timmer [1985] (RBT) give a survey of results from this literature that are directly relevant to finding the solution to (4.2), as well as describing a new procedure, "multi-level single linkage" which appears to provide performance superior to a variety of now standard methods. Before describing this technique, however, we first consider two methods for solving (4.2) that are relatively familiar to the neural computation community: the method of simulated annealing and the genetic algorithm. Both of these methods for function optimization have been applied in the present context or in related contexts. Because of their relative familiarity,

we shall not go into great detail regarding the specifics of implementation, but indicate general features of these methods.

The method of simulated annealing proceeds by viewing $\hat{\lambda}_n$ as giving an "energy landscape" over the state space W. It is desired to settle into a low energy state, the lowest being $\hat{w}_n$. Different annealing strategies arise depending upon whether W is a finite set or is a continuum, but the basic idea is to start at some initial weight vector and compute the "energy" (value of $\hat{\lambda}_n$) for a nearby weight vector. If energy is lower, move to the new vector. If energy is higher, move to the new vector with a probability controlled by the annealing "temperature" schedule. By setting the temperature high initially, one may escape from local minima. The "temperature" is lowered at an appropriate rate so as to control the probability of jumping away from relatively good minima. Hajek [1985, 1988] gives a useful survey and some theorems establishing conditions under which simulated annealing ultimately delivers the solution $\hat{w}_n$ to (4.2). See also Davis [1987].

It is useful to recognize that such procedures leave us twice removed from the optimal weights w^*, in a certain sense. If we could find $\hat{w}_n$, we would be once removed, effectively by sampling variation, although the results described above show that this sampling variation gets averaged out as n becomes large. However, finding $\hat{w}_n$ is only guaranteed in the limit of the annealing process. Because the annealing process must be terminated at some finite time, we are once removed from $\hat{w}_n$, and therefore twice removed from w^*. The most we can hope for is that weights, say $\ddot{w}_n$, delivered by annealing after some finite time, will be close to $\hat{w}_n$ in the sense that $\hat{\lambda}_n(\ddot{w}_n)$ is close to $\hat{\lambda}_n(\hat{w}_n)$. These weights could be far apart in standard metrics on W, but this is not of

major concern: our primary concern is with measured average performance.

The statistical properties of $\ddot{w}_n$ are not necessarily identical to those of $\hat{w}_n$. However, if $\ddot{w}_n$ delivers a local minimum of $\hat{\lambda}_n$ we can view $\ddot{w}_n$ as minimizing $\hat{\lambda}_n$ over some restriction of W, and regain similar statistical properties with respect to this restriction. We discuss finding a local minimum in more detail below.

The genetic algorithm (Holland [1975]) proceeds by viewing the opposite of $\hat{\lambda}_n$, i.e. $-\hat{\lambda}_n$, as a fitness function and w as a "DNA vector." Use of the genetic algorithm for function optimization is treated by Goldberg [1989] (see also Davis [1987]). The basic idea is to begin with a population of N "individuals" with "DNA" $w^i, i = 1, ..., N$. The fitness of each individual is evaluated as $-\hat{\lambda}_n(w^i)$. Individuals mate with other individuals, exchanging "genetic material" in a manner bearing certain analogies to the exchange of DNA in biological organisms. More fit individuals are more likely to mate; further, the exchange of "DNA" is governed by "genetic operators" such as "cross-over" and "mutation" that allow for local search (mutation) and distant search (cross-over) in W. The result is a new generation of individuals with new "DNA." The process continues for many generations. Heuristically, one might expect the optimal individual $\hat{w}_n$ to emerge from this process as the number of generations becomes large.

To date, there does not appear to be a theoretical result guaranteeing that $\hat{w}_n$ is indeed produced in the limit; such a result would be highly desirable. Nevertheless, the method does seem to perform reasonably well in applications. Typically, the method delivers weights in the neighborhood of an optimum relatively quickly, but can be very slow to find the optimum itself, owing to its simple method of local search. To aid the production of a highly fit individual in the present context, it is desirable to "clone" the

most fit individual from one generation to the next. Also, it appears desirable to treat the elements of w as distinct entities in the cross-over process, with attention also paid to keeping together "clumps" of hidden units, as these typically collectively encode information leading to good fitness.

Again, weights produced by the genetic algorithm are twice removed from w^* for the same reasons as with the method of simulated annealing. The identical comments apply, especially as regards obtaining a local minimum of $\hat{\lambda}_n$ for the purpose of exploiting the statistical properties of the resulting weights.

This discussion of particular learning methods clarifies the relationship between two separate areas relevant for the analytic investigation of network learning. The first is the area of statistical analysis, which allows us to study the properties of any procedure that delivers a solution to (4.2). These properties are fairly well established. The second is the area of optimization theory, which delivers methods leading to the solution of (4.2). Such methods present a current challenge; the vast literature of optimization theory can be expected to yield a variety of useful methods for attempting to solve (4.2) in specific applications.

As an example, we describe the multi-level single linkage algorithm of Rinnooy Kan, Boender and Timmer [1985]. This method is a variant of the "multi-start" method, which has three steps:

1.) Draw a weight vector w from the uniform distribution over W.

2.) Carry out a local search starting from w (see below for methods of local search) to obtain a local minimizer $\ddot{w}_n$, say.

3.) If $\hat{\lambda}_n(\ddot{w}_n)$ is the smallest value obtained so far, put $\hat{w}_n = \ddot{w}_n$. Return to step 1.

The procedure continues until a stopping criterion is met.

The multi-level single linkage technique is designed to improve the efficiency of the multi-start procedure by performing local search for a minimum, starting not from every point drawn in step 1, but only from points satisfying certain criteria. Specifically, draw a sample of weight vectors $w^i, i = 1, ..., N$ from the uniform distribution over W and initiate local search from each weight, unless: (1) w^i is too close to the boundary of W (within a distance $\tau > 0$, in RBT's notation); (2) w^i is too close to a previously identified local minimizer (within a distance $\upsilon > 0$ in RBT's notation); or (3) there is a weight vector $w^j, j \neq i$, such that $\hat{\lambda}_n(w^j) < \hat{\lambda}_n(w^i)$ and w^j is close to w^i (within a distance $r_N > 0$ in RBT's notation). Timmer [1984] proves that if r_N is chosen appropriately and tends to zero as $N \to \infty$, then any local minimum $\ddot{w}_n$ (and consequently global minimum $\hat{w}_n$) will be found within a finite number of iterations, with probability 1. The reader is referred to Timmer [1984] for further discussion.

Methods capable of solving (4.2) locally are themselves the subject of a voluminous literature. We merely sketch the outline of some gradient descent techniques that are straightforward to implement. Specifically, if $\hat{\lambda}_n$ is differentiable in w, an iteration can be constructed as

$$\tilde{w}_n^{(k+1)} = \tilde{w}_n^{(k)} - \eta_k H_n^{(k)} \nabla \hat{\lambda}_n(\tilde{w}_n^{(k)}) \quad k = 0, 1, 2, ...$$

where $\tilde{w}_n^{(k)}$ is the estimate at the kth iteration, $\tilde{w}_n^{(0)}$ is any starting value, η_k is a positive step-size parameter, $H_n^{(k)}$ is an $s \times s$ positive definite matrix and ∇ is the gradient operator with respect to w, so that $\nabla \hat{\lambda}_n$ is an $s \times 1$ vector. Different choices for η_k and $H_n^{(k)}$ implement different specific gradient descent methods. A discussion of these and much

additional relevant material can be found in Ortega and Rheinboldt [1970] and Rheinboldt [1974]. Note that the sometimes extreme local irregularity ("roughness," "ruggedness") of the function $\hat{\lambda}_n$ over W arising in network learning applications may require development and use of appropriate modifications to the standard methods.

Under appropriate regularity conditions, $\tilde{w}_n^{(k)}$ converges as $k \to \infty$ to $\tilde{w}_n$, a vector such that $\nabla \hat{\lambda}_n(\tilde{w}_n) = 0$. These equations are the necessary first order conditions for a local minimum of $\hat{\lambda}_n$ interior to W.

Under appropriate conditions, it can be further shown that $\tilde{w}_n$ tends to w^*, a parameter vector solving the problem $\nabla \lambda(w) = 0$. When interchange of derivative and integral is possible, we have

$$\begin{aligned} \nabla \lambda(w) &= \nabla \int l(z, w) \nu(dz) \\ &= \int \nabla\, l(z, w) \nu(dz) \\ &= E(\nabla\, l(Z, w)). \end{aligned}$$

Thus, seeking a solution to $\nabla \lambda(w) = 0$ is the same as seeking a solution to the problem

$$E(\nabla\, l(Z, w)) = 0. \tag{4.3}$$

Because ν is unknown, we cannot solve this problem directly. Nevertheless, the gradient descent methods just discussed provide one approach to attempting to find such a solution using available sample information, because $\nabla \hat{\lambda}_n(w) = n^{-1} \sum_{t=1}^{n} \nabla\, l(Z_t, w)$.

4.b Learning by Recursive Methods

In 1951, Robbins and Monro [1951] considered the problem of finding a solution to the problem

$$E(m(Z,w)) = \int m(z,w)\nu(dz) = 0, \tag{4.4}$$

when the expectation cannot be computed because ν is unknown. Instead, error laden observations on $E(m(Z,w))$ are given by realizations of the random variables $m(Z_t,w)$. Robbins and Monro [1951] proposed the method of "stochastic approximation" for finding an approximate solution to (4.4) using the recursion

$$\tilde{w}_n = \tilde{w}_{n-1} - \eta_n m(Z_n, \tilde{w}_{n-1}), \quad n = 1,2,..., \tag{4.5}$$

where $\tilde{w}_0$ is arbitrary and η_n is a learning rate. Robbins and Monro [1951] studied their procedure for the case in which w is a scalar; Blum [1954] extended their analysis to the vector case.

By setting $m(z,w) = \nabla l(z,w)$, we can apply the Robbins-Monro procedure to obtain an approximate solution to the problem (4.3). The recursion is simply

$$\tilde{w}_n = \tilde{w}_{n-1} - \eta_n \nabla l(Z_n, \tilde{w}_{n-1}), \quad n = 1,2,.... \tag{4.6}$$

When $\pi(y,o) = (y-o)^2/2$, we have

$$\nabla l(z,w) = -\nabla f(x,w)(y - f(x,w))$$

so that

$$\tilde{w}_n = \tilde{w}_{n-1} + \eta_n \nabla f(X_n, \tilde{w}_{n-1})(Y_n - f(X_n, \tilde{w}_{n-1})), \quad n = 1,2,....$$

This is easily recognized as the method of back-propagation (Werbos [1974], Parker [1982], Rumelhart, Hinton and Williams [1986]). Thus, the method of back-propagation can be viewed as an application of the Robbins-Monro [1951] stochastic approximation procedure to solving the first order conditions for a nonlinear least squares regression problem.

The statistical concepts of consistency and limiting distribution are immediately relevant for studying the behavior of $\{\tilde{w}_n\}$. Results in the statistics and systems identification literature can be applied directly to investigate the properties of $\{\tilde{w}_n\}$. White [1989b] applies results of Ljung [1977] and Walk [1977] to obtain consistency and limiting distribution results for the method of back-propagation as well as for the recursion (4.5). In these results, the random variables $Z_1, Z_2, \ldots$ are assumed to be statistically independent. Such an assumption is implausible for the analysis of time series data, so Kuan and White [1989] apply results of Kushner and Clark [1978] and Kushner and Huang [1979] to establish consistency and limiting distribution results for dependent sequences of random variables. An interesting feature of these results is that they specify conditions on the learning rate η_n that are *necessary* to ensure the desired convergence results. In particular, the most rapid convergence occurs when $\eta_n \leq \Delta\, n^{-1}$ for some $\Delta < \infty$.

The limiting distribution results of White [1989b] and of Kuan and White [1989] can be used to test the irrelevant input hypothesis and the irrelevant hidden unit hypothesis in ways analogous to those discussed earlier. Also of interest is the fact that the recursion (4.5) can be used to generate modifications of the method of back-propagation that have improved convergence and statistical efficiency properties. Several of these are described by Kuan and White [1989].

A fundamental limitation of the recursion (4.6) is that it is not guaranteed to converge stochastically, and if it does converge, it generally will not converge to a global solution of (4.1). Under favorable conditions, it may converge to a local solution to (4.1). Kushner [1987] has studied a modification of (4.6) guaranteed to converge *w.p.*1 to a

global solution to (4.1) as $n \to \infty$. His method embodies a form of annealing. The recursion is

$$\tilde{w}_n = \tilde{w}_{n-1} + \eta_n (\nabla l(Z_n, \tilde{w}_{n-1}) + \zeta_n)$$

where $\{\zeta_n\}$ is a sequence of independent identically distributed Gaussian random variables. Convergence to a global optimum as $n \to \infty$ occurs almost surely, provided that η_n is proportional to $1 / \log n$. This gives very slow convergence.

The discussion of this section has so far related a variety of well-known learning methods for artificial neural networks to existing methods of statistical estimation. The statistics literature suggests a variety of additional relevant procedures that to my knowledge have not yet been proposed as network learning methods. One such procedure is that of Kiefer and Wolfowitz [1952]. The Kiefer-Wolfowitz procedure and its variants are useful for situations in which computing ∇l is difficult or impossible. Instead of using ∇l, use is made of an estimate of ∇l based on observations on l.

A particularly convenient variant of the Kiefer-Wolfowitz procedure known as the "method of random directions" has been analyzed by Kushner and Clark [1978]. To implement this method, one chooses a sequence of real constants $\{c_n\}$ and a sequence of direction vectors $\{d_n\}$ uniformly distributed over the unit sphere in R^s. Weights are generated by the recursion

$$\tilde{w}_n = \tilde{w}_{n-1} - \eta_n d_n (l(Z_n, \tilde{w}_{n-1} + c_n d_n) - l(Z_n, \tilde{w}_{n-1} - c_n d_n)) / 2c_n, \quad n = 1, 2, \ldots. \tag{4.7}$$

The term $d_n (l(Z_n, \tilde{w}_{n-1} + c_n d_n) - l(Z_n, \tilde{w}_n - c_n d_n)) / 2c_n$ plays the role of $\nabla l(Z_n, \tilde{w}_{n-1})$ in the Robbins-Monro procedure (4.6). Kushner and Clark [1978] give conditions under which $\tilde{w}_n \to w^t$ *a.s.*$-P$, where w^t is now a local solution of the problem (4.1).

By setting $l(z, w) = (y - f(x, w))^2 / 2$ in (4.7) we obtain a version of back-propagation that requires no computation of the gradient of the network output function, ∇f. Instead, we rely on a random local exploration of the loss function. This may prove convenient for training multilayer networks with a large number of hidden layers, as the effort required to compute the gradient for back-propagation can be large in these cases.

A useful review of recursive estimation methods such as the Robbins-Monro and Kiefer-Wolfowitz procedures has recently been given by Ruppert [1989]. Much of the material contained there has direct relevance for learning in artificial neural networks.

4.c Summary

To summarize this section, a large class of learning methods for artificial neural networks can be viewed as statistical procedures for solving the problems (4.1) or (4.3). Concepts of stochastic convergence provide an appropriate framework in which to analyze the properties of these procedures. Existing results in the statistics, econometrics, systems identification and optimization theory literatures can be applied directly to describe the properties of network learning methods. These properties can be exploited to answer questions about optimal network architectures using the tools of statistical inference. Further, existing methods can suggest useful and novel network learning procedures, such as the multi-level single linkage algorithm or the Kiefer-Wolfowitz approach to back-propagation.

5. NONPARAMETRIC ESTIMATION WITH FEEDFORWARD NETWORKS

In all of the foregoing discussion, we have considered learning methods for networks of fixed complexity. Despite the great flexibility that such networks can afford

in their input-output response (e.g. in their ability to approximate arbitrary mappings), they are nevertheless fundamentally limited. In particular, feedforward networks of fixed complexity will be able to provide only partial approximations to arbitrary mappings; their performance for especially complicated mappings can be quite poor. However, it is now well established that hidden layer feedforward networks with as few as a single hidden layer are capable of arbitrarily accurate approximation to an arbitrary mapping provided that sufficiently many hidden units are available (see Carroll and Dickinson [1989], Cybenko [1989], Funahashi [1989], Hecht-Nielsen [1989], Hornik, Stinchcombe and White [1989a,b] (HSW) and Stinchcombe and White [1989]). It is natural to ask whether it is possible to devise a learning procedure that can learn an arbitrarily accurate approximation to an arbitrary mapping. In this section, we review some recent results showing that this is indeed possible. These results are obtained by permitting the complexity of the network to grow at an appropriate rate relative to the size of the available training set.

For concreteness, we assume that our interest centers on learning the conditional expectation function, which we now denote θ_o, so that $\theta_o(X) = E(Y \mid X)$. (We have θ_o corresponding to g in our previous notation.) Other aspects of the conditional distribution γ of Y given X can be given a similar treatment.

Because in practice we always have a training set of finite size n and because θ_o is an element of a space of functions (say Θ) and is generally not an element of a finite dimensional space, we have essentially no hope of learning θ_o in any complete sense from a sample of fixed finite size. Nevertheless, it is possible to approximate or estimate θ_o to some degree of accuracy using a sample of size n, and to construct increasingly accurate approximations with increasing n. Let a learned approximation to θ_o based on a

training set of size n be denoted $\hat{\theta}_n$. Just as in our discussion of the convergence properties of learned network weights $\hat{w}_n$, we can define appropriate notions of stochastic convergence for learned approximations $\hat{\theta}_n$, and it is in terms of this stochastic convergence that the approximations may become increasingly accurate.

To define the appropriate notions of stochastic convergence, we need a way to measure distances between different functions belonging to Θ. A formal way to do this is to introduce a "metric" ρ, that is, a real-valued function on $\Theta \times \Theta$ which has the properties that $\rho(\theta_1, \theta_2) \geq 0$ (non-negativity), $\rho(\theta_1, \theta_2) = \rho(\theta_2, \theta_1)$ (symmetry) and $\rho(\theta_1, \theta_2) \leq \rho(\theta_1, \theta_3) + \rho(\theta_3, \theta_2)$ (triangle inequality) for all $\theta_1, \theta_2, \theta_3$ in Θ. When $\rho(\theta_1, \theta_2) = 0$, we view θ_1 and θ_2 as identical. The pair (Θ, ρ) is known as a "metric space." For any function space Θ there are usually many different possible choices for ρ. However, once a suitable metric is specified, we can define stochastic convergence in terms of the chosen metric. The property of strong (ρ–) consistency of $\hat{\theta}_n$ for θ_o holds when $\rho(\hat{\theta}_n, \theta_o) \rightarrow 0$ (as $n \rightarrow \infty$) *a.s.–P*. The property of weak (ρ–) consistency of $\hat{\theta}_n$ for θ_o holds when $\rho(\hat{\theta}_n, \theta_o) \rightarrow 0$ *prob–P*. Because weak consistency is often easier to establish, we focus only on weak consistency, and drop the explicit use of the word "weak."

In a very precise sense, then, a "consistent" learning procedure for θ_o is one capable of generating a sequence of approximations $\hat{\theta}_n$ to θ_o having the property that $\rho(\hat{\theta}_n, \theta_o) \rightarrow 0$ *prob–P*, and it is in this sense that such approximations can approximate an arbitrary function θ_o arbitrarily well. Equivalently, the probability that $\hat{\theta}_n$ exceeds any specified level of approximation error relative to θ_o as measured by the metric ρ tends to zero as the sample size n tends to infinity. Procedures that are not consistent will always

make errors in classification, recognition, forecasting, or pattern completion (forms of generalization) that are eventually avoided by a consistent procedure. The only errors ultimately made by a consistent procedure are the inherently unavoidable errors ($\varepsilon = Y - \theta_o(X)$) arising from any fundamental randomness or fuzziness in the true relation between X and Y.

White [1988] uses statistical theory for the "method of sieves" (Grenander [1981], Geman and Hwang [1982], White and Wooldridge [1989]) to establish that multilayer feedforward networks can be used to obtain a consistent learning procedure for θ_o under fairly general conditions. The method of sieves is a general approach to nonparametric estimation in which an object of interest θ_o lying in a general (i.e. not necessarily finite dimensional) space Θ is approximated using a sequence of parametric models in which dimensionality of the parameter space grows along with the sample size. The success of this approach requires the approximating parametric models to be capable of arbitrarily accurate approximation to elements of Θ as the underlying parameter space grows. For this reason, Fourier series (e.g. Gallant and Nychka [1987]) and spline functions (e.g. Wahba [1984], Cox [1984]) are commonly used in this context. Because multilayer feedforward networks have universal approximation properties, they are also suitable for such use. Without the universal approximation property, attempts at nonparametric estimation using feedforward networks would be doomed from the outset.

White [1988] considers approximations obtained using single hidden layer feedforward networks with output functions

$$f^q(x, w^q) \equiv w_{10} + \sum_{j=1}^{q} w_{1j}\psi(\tilde{x}'w_{0j}),$$

where $w^q \equiv (w_0', w_1')'$ is the $s \times 1$ ($s = q(r + 2) + 1$) vector of network weights. There are q

hidden units. The vector w_0 contains the input to hidden unit weights, $w_0 \equiv (w_{01}', \ldots, w_{0q}')'$, $w_{0j} \equiv (w_{0j0}, w_{0j1}, \ldots, w_{0jr})'$, and the vector w_1 contains the hidden to output weights $w_1 \equiv (w_{10}, \ldots, w_{1q})$. ψ is the hidden unit activation function, and $\tilde{x} = (1, x')'$. Note that the network output function and the weight vector are explicitly indexed by the number of hidden units, q.

Because the complexity of such networks is indexed solely by q, we construct a sequence of approximations to θ_o by letting q grow with n at an appropriate rate, and for given n (hence given q) selecting connection strengths $\hat{w}_n$ so that $\hat{\theta}_n \equiv f^{q_n}(\cdot, \hat{w}_n)$ provides an approximation to the unknown regression function θ_o that is the best possible in an appropriate sense, given the sample information.

To formulate precisely a solution to the problem of finding $\hat{\theta}_n$, White defines the set

$$T(\psi, q, \Delta) \equiv \{\theta \in \Theta : \theta(\cdot) = f^q(\cdot, w^q), \sum_{j=0}^{q} | w_{1j} | \leq \Delta, \sum_{j=1}^{q} \sum_{i=0}^{r} | w_{0ji} | \leq q\Delta\}.$$

This is the set of all single hidden layer feedforward networks with q hidden units having activation functions ψ, and with connection strengths satisfying a particular restriction on their sum norm, indexed by Δ. This last restriction arises from certain technical aspects of the analysis. A sequence of "sieves" $\{\Theta_n(\psi)\}$ is constructed by specifying sequences $\{q_n\}$ and $\{\Delta_n\}$ and setting $\Theta_n(\psi) = T(\psi, q_n, \Delta_n)$, $n = 1, 2, \ldots$. The sieve $\Theta_n(\psi)$ becomes finer (less escapes) as $q_n \to \infty$ and $\Delta_n \to \infty$. For given sequences $\{q_n\}$ and $\{\Delta_n\}$, the "connectionist sieve estimator" $\hat{\theta}_n$ is defined as a solution to the least squares problem (appropriate for learning $E(Y \mid X)$)

$$\min_{\theta \in \Theta_n(\psi)} n^{-1} \sum_{t=1}^{n} [Y_t - \theta(X_t)]^2 / 2, \quad n = 1, 2, \ldots. \tag{5.1}$$

Associated with $\hat{\theta}_n$ is an estimator $\hat{w}_n$ of dimension $s_n \times 1$ ($s_n = q_n(r + 2) + 1$) such that

$\hat{\theta}_n(\cdot) = f^{q_n}(\cdot, \hat{w}_n)$. The estimator $\hat{w}_n$ is defined as the solution to the problem

$$\min_{w^{q_n} \in W_n} n^{-1} \sum_{t=1}^{n} [Y_t - f^{q_n}(X_t, w^{q_n})]^2 / 2, \tag{5.2}$$

where $W_n = \{w^{q_n} : \sum_{j=0}^{q_n} | w_{1j} | \leq \Delta_n, \sum_{j=1}^{q_n} \sum_{i=1}^{r} | w_{0ji} | \leq q_n \Delta_n \}$. Comparing this to the problem (4.3), we see that the only difference is that (5.2) explicitly references network complexity which is now a function of the size n of the available body of empirical evidence.

White [1988] gives precise conditions on Θ, $\{Z_t\}$ (equivalently, P), ψ, $\{q_n\}$ and $\{\Delta_n\}$ that ensure the consistency of $\hat{\theta}_n$ for θ_o in Θ in the sense of the root mean square metric ρ_2,

$$\rho_2(\theta_1, \theta_2) = (\int [\theta_1 - \theta_2]^2 d\mu)^{1/2}.$$

Note that the integral is taken with respect to the environment measure μ. White's conditions are straightforward to describe. Θ is taken to be the space of square integrable functions on a given compact subset of $\mathbb{R}^r$ ($\Theta = \{\theta : K \to \mathbb{R} : \int_K \theta^2 d\mu < \infty$, K a compact subset of $\mathbb{R}^r\}$). The probability measure P is assumed to generate identically distributed random variables Z_t having joint probability measure ν, with subvector X_t having probability measure μ such that $\mu(K) = 1$. For simplicity, Y_t is also assumed bounded, although this condition can be relaxed. The probability measure P also governs the interdependence of Z_t and Z_τ, $t \neq \tau$. White considers the case of independent random variables (appropriate for cross-section samples) and the case of "mixing" random variables (appropriate for time-series samples). The activation functions ψ are chosen to be any activation function that permits single hidden layer feedforward networks to possess the universal approximation property in (Θ, ρ_2). For example, ψ can be sigmoid,

as shown by HSW.

With these conditions specified, it is possible to derive growth rates for network complexity ensuring that $\rho_2(\hat{\theta}_n, \theta_o) \to 0\ prob\text{-}P$, i.e. that single hidden layer feedforward networks are capable of learning an arbitrarily accurate approximation to an unknown function, provided that they increase in complexity at an appropriate rate.

In fact, the theoretical results require proper control of both Δ_n and q_n. The appropriate choice for Δ_n is proven to be such that $\Delta_n \to \infty$ as $n \to \infty$ and $\Delta_n = o(n^{1/2})$, i.e., $n^{-1/2}\Delta_n \to 0$. A standard choice in the sieve estimation literature is $\Delta_n \propto \log n$. The appropriate choice for q_n depends on Δ_n and on the assumed dependence properties of $\{Z_t\}$. When $\{Z_t\}$ is an independent sequence, it suffices that $q_n \to \infty$ and $q_n \Delta_n^4 \log(q_n \Delta_n) = o(n)$; when $\{Z_t\}$ is a mixing sequence, it suffices that $q_n \Delta_n^2 \log(q_n \Delta_n) = o(n^{1/2})$. For the choice $\Delta_n \propto \log n$, these conditions permit $q_n \propto n^{1-\delta}, 0 < \delta < 1$, for the independent case and $q_n \propto n^{(1-\delta)/2}$ for the mixing case. The underlying justification for these growth rates is quite technical and cannot be given a meaningful simple explanation; most of the theoretical analysis is devoted to obtaining these rates. Nevertheless, their purpose is clear: they serve to prevent network complexity from growing so fast that overfitting results in the limit.

These analytical results show only that network learning of an arbitrarily accurate approximation of an arbitrary mapping is possible. They do not provide more than very general guidance on how this can be done, and what guidance they do provide suggests that such learning will be hard. In particular, the learning method requires solution of (5.2). Global optimization methods, such as those discussed in Section 4.a(iv) are appropriate.

Furthermore, although these results do provide asymptotic guidelines on growth of network complexity, they say nothing about how to determine adequate network complexity in any specific application with a given training set of size n. The search for an appropriate technique for determining network complexity has been the focus of considerable effort to date (e.g. Rumelhart [1988], Ash [1989], Hirose, Yamashita and Hijiya [1989]). It is apparent that methods developed by statisticians will prove helpful in this search. In particular, White [1988] discusses use of the method of cross-validation (e.g. Stone [1974]) to determine network complexity appropriate for a training set of given size.

White's analysis does not treat the limiting distribution of $\hat{\theta}_n$. This analysis is more difficult than that associated with $\hat{w}_n$ because $\hat{\theta}_n$ has a probability distribution over a function space. A study of this distribution is of theoretical interest, and may also be of some practical use. Results of Andrews [1988] may be applicable to obtain the limiting distributions of linear and nonlinear functionals of $\hat{\theta}_n$. These would allow construction of asymptotic confidence intervals for the value of θ_o at a given point x_o, for example.

Also of interest are hypothesis tests that will permit inference about the nature of a given mapping of interest. Specifically, some theory of the phenomenon of interest might suggest that a particular unknown mapping θ_o has a specific linear or nonlinear form, so that one might formulate the null hypothesis $H_o : \theta_o \in \Theta_o$, where Θ_o is some specific class of functions having the specified property (e.g. affine functions or some specified parametric family). The alternative is that θ_o does not belong to Θ_o, i.e. $H_a : \theta \notin \Theta_o$. Tests of this H_o against H_a have been extensively studied in the econometrics literature, where they are known as specification tests. A specification test using single hidden

layer feedforward networks has been proposed by White [1989c] and investigated by Lee, White and Granger [1989]. Most such specification tests are "blind" to certain alternatives, however, in that they will fail to detect certain departures from H_o no matter how large is n. Recent work of Wooldridge [1989] has exploited the nonparametric estimation capabilities of series estimators, a special class of sieve estimators, to obtain specification tests that are "consistent," meaning that they can detect any departure from H_o with probability approaching 1 as n becomes large. It is plausible that Wooldridge's approach can be applied to the connectionist sieve estimator as well, so that consistent tests of H_o can be obtained using feedforward networks.

SUMMARY AND CONCLUDING REMARKS

It is the premise of this review that learning methods in artificial neural networks are sophisticated statistical procedures and that tools developed for the study of statistical procedures generally can not only yield useful insights into the properties of specific learning procedures but also suggest valuable improvements in, alternatives to and generalizations of existing learning procedures. Particularly applicable are asymptotic analytical methods that describe the behavior of statistics when the size n of the training set is large. The study of the stochastic convergence properties (consistency, limiting distribution) of any proposed new learning procedure is strongly recommended, in order to determine what it is that the network eventually learns and under what specific conditions. Derivation of the limiting distribution will generally reveal the statistical efficiency of the new procedure relative to existing procedures and may suggest modifications capable of improving statistical efficiency. Furthermore, the availability of the limiting distribution makes possible valid statistical inferences. Such inferences can

be of great value in the investigation of optimal network architectures in particular applications. A wealth of applicable theory is already available in the statistics, econometrics, systems identification and optimization theory literatures.

Among the applications of results already available in these literatures are some potentially useful learning methods for artificial neural networks based on the multi-level single linkage and the Kiefer-Wolfowitz procedures, as well as a demonstration of the usefulness of multilayer feedforward networks for nonparametric estimation of an unknown mapping. We have described recent work of White [1988] along these lines, establishing that arbitrary mappings can indeed be learned using multilayer feedforward networks.

It is also evident that the field of statistics has much to gain from the connectionist literature. Analyzing neural network learning procedures poses a host of interesting theoretical and practical challenges for statistical method; all is not cut and dried. Most importantly however, neural network models provide a novel, elegant and extremely rich class of mathematical tools for data analysis. Application of neural network models to new and existing datasets holds the potential for fundamental advances in empirical understanding across a broad spectrum of the sciences. To realize these advances, statistics and neural network modeling must work together, hand in hand.

REFERENCES

Andrews, D.W.K. [1988]: "Asymptotic Normality of Series Estimators for Various Nonparametric and Semi-parametric Estimators," Yale University, Cowles Foundation Discussion Paper 874.

Ash, T. [1989]: "Dynamic Node Creation in Backpropagation Networks," Poster presentation, International Joint Conference on Neural Networks, Washington, D.C.

Blum, J. [1954]: "Multivariate Stochastic Approximation Methods," *Annals of Mathematical Statistics* 25, 737-744.

Carroll, S.M. and B.W. Dickinson [1989]: "Construction of Neural Nets Using the Radon Transform," in *Proceedings of the International Joint Conference on Neural Networks*, Washington, D.C. San Diego: SOS Printing, pp. I:607-611.

Cox, D. [1984]: "Multivariate Smoothing Splines," *SIAM Journal of Numerical Analysis* 21, 789-813.

Cybenko, G. [1989]: "Approximation by Superpositions of a Sigmoidal Function," *Mathematics of Control, Signals and Systems*, forthcoming.

Domowitz, I. and H. White [1982]: "Misspecified Models with Dependent Observations," *Journal of Econometrics* 20, 35-50.

Davies, R.B. [1977]: "Hypothesis Testing When a Nuisance Parameter is Present Only Under the Alternative," *Biometrika* 64, 247-254.

Davies, R.B. [1987]: "Hypothesis Testing When a Nuisance Parameter is Present Only Under the Alternative," *Biometrika* 74, 33-43.

Davis, L. (ed.) [1987]. *Genetic Algorithms and Simulated Annealing*. London: Pitman.

Funahashi, K. [1989]: "On the Approximate Realization of Continuous Mappings by Neural Networks," *Neural Networks* 2, 183-192.

Gallant, A.R. and D. Nychka [1987]: "Semi-nonparametric Maximum Likelihood Estimation," *Econometrica* 55, 363-390.

Geman, S. and C. Hwang [1982]: "Nonparametric Maximum Likelihood Estimation by the Method of Sieves," *Annals of Statistics* 10, 401-414.

Goldberg, D. [1989]. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading MA: Addison-Wesley.

Grenander, U. [1981]. *Abstract Inference*. New York: Wiley.

Hajek, B. [1985]: "A Tutorial Survey of Theory and Applications of Simulated Annealing," in *Proceedings of the 24th IEEE Conference on Decision and Control*, pp. 755-760.

Hajek, B. [1988]: "Cooling Schedules for Optimal Annealing," *Mathematics of Operations Research* 13, 311-329.

Hecht-Nielsen, R. [1989]: "Theory of the Back-Propagation Neural Network," in *Proceedings of the International Joint Conference on Neural Networks*, Washington D.C. San Diego: SOS Printing, pp. I:593-606.

Hirose, Y., K. Yamashita and S. Hijiya [1989]: "Back-Propagation Algorithm Which Varies the Number of Hidden Units," Poster presentation, International Joint Conference on Neural Networks, Washington, D.C.

Holland, J. [1975]. *Adaptation in Natural and Artificial Systems.* Ann Arbor: University of Michigan Press.

Hornik, K., M. Stinchcombe and H. White [1989a]: "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks* 2, forthcoming.

Hornik, K., M. Stinchcombe and H. White [1989b]: "Universal Approximation of an Unknown Mapping and Its Derivatives Using Multilayer Feedforward Networks," UCSD Department of Economics Discussion Paper.

Kiefer, J. and J. Wolfowitz [1952]: "Stochastic Estimation of the Maximum of a Regression Function," *Annals of Mathematical Statistics* 23, 462-466.

Kuan, C.-M. and H. White [1989]: "Recursive M-Estimation, Nonlinear Regression and Neural Network Learning with Dependent Observations," UCSD Department of Economics Discussion Paper.

Kullback, S. and R.A. Leibler [1951]: "On Information and Sufficiency," *Annals of Mathematical Statistics* 22, 79-86.

Kushner, H. [1987]: "Asymptotic Global Behavior for Stochastic Approximations and Diffusions with Slowly Decreasing Noise Effects: Global Minimization via Monte Carlo," *SIAM Journal on Applied Mathematics* 47, 169-185.

Kushner, H. and D. Clark [1978]. *Stochastic Approximation Methods for Constrained and Unconstrained Systems.* Berlin: Springer-Verlag.

Kushner, H. and H. Huang [1979]: "Rates of Convergence for Stochastic Approximation Type Algorithms," *SIAM Journal of Control and Optimization* 17, 607-617.

Lee, T.-H., H. White and C.W.J. Granger [1989]: "Testing for Neglected Nonlinearity in Time Series Models: A Comparison of Neural Network Methods and Alternative Tests," UCSD Department of Economics Discussion Paper.

Ljung, L. [1977]: "Analysis of Recursive Stochastic Algorithms," *IEEE Transactions on Automatic Control* AC-22, 551-575.

Ortega, J. and W. Rheinboldt [1970]. *Iterative Solution of Nonlinear Equations in Several Variables*. New York: Academic Press.

Parker, D.B. [1982]: "Learning Logic," Invention Report 581-64, File 1, Office of Technology Licensing, Stanford University.

Renyi, A. [1961]: "On Measures of Entropy and Information," in *Proceedings of the Fourth Berkeley Symposium in Mathematical Statistics*, Vol 1. Berkeley: University of California Press, pp. 547-561.

Rheinboldt, W. [1974]. *Methods for Solving Systems of Nonlinear Equations*. Philadelphia: SIAM.

Rinnooy Kan, A.H.G., C.G.E. Boender and G. Th. Timmer [1985]: "A Stochastic Approach to Global Optimization," in K. Schittkowski, ed., *Computational Mathematical Programming*, NATO ASI Series, Vol F15. Berlin: Springer-Verlag, pp. 281-308.

Robbins, H. and S. Monro [1951]: "A Stochastic Approximation Method," *Annals of Mathematical Statistics* 22, 400-407.

Rumelhart, D. [1988]: "Parallel Distributed Processing," Plenary Lecture, *IEEE International Conference on Neural Networks*, San Diego.

Rumelhart, D.E., G.E. Hinton and R.J. Williams [1986]: "Learning Internal Representations by Error Propagation," in D.E. Rumelhart and J.L. McClelland eds., *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, Vol 1. Cambridge: MIT Press, pp. 318-362.

Ruppert, D. [1989]: "Stochastic Approximation," in B. Ghosh and P. Sen eds., *Handbook of Sequential Analysis*. New York: Marcel Dekker, forthcoming.

Stinchcombe, M. and H. White [1989]: "Universal Approximation Using Feedforward Networks with Non-Sigmoid Hidden Layer Activation Functions," in *Proceedings of the International Joint Conference on Neural Networks*, Washington, D.C. San Diego: SOS Printing, pp. I:613-617.

Stone, M. [1974]: "Cross-Validatory Choice and Assessment of Statistical Predictions," *Journal of the Royal Statistical Society* Series B 36, 111-133.

Timmer, G. Th. [1984]: "Global Optimization: A Stochastic Approach," unpublished Ph.D. Dissertation, Erasmus Universiteit Rotterdam, Centrum voor Wiskunde en Informatica.

Wahba, G. [1984]: "Cross-Validated Spline Methods for the Estimation of Multivariate Functions from Data on Functionals," in H.A. David and H.T. David eds., *Statistics: An Appraisal*. Ames, Iowa: Iowa State University Press, pp. 205-235.

Walk, H. [1977]: "An Invariance Principle for the Robbins-Monro Process in a Hilbert Space," *Zeitschrift für Warscheinlichskeitstheorie und Verwandete Gebiete* 39, 135-150.

Werbos, P. [1974]: "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences," unpublished Ph.D. Dissertation, Harvard University, Department of Applied Mathematics.

White, H. [1981]: "Consequences and Detection of Misspecified Nonlinear Regression Models," *Journal of the American Statistical Association*, 76, 419-433.

White, H. [1982]: "Maximum Likelihood Estimation of Misspecified Models," *Econometrica* 50, 1-25.

White, H. [1984a]: "Maximum Likelihood Estimation of Misspecified Dynamic Models," in T. Dijkstra ed., *Misspecification Analysis*. New York: Springer-Verlag, pp. 1-19.

White, H. [1984b]. *Asymptotic Theory for Econometricians*. New York: Academic Press.

White, H. [1988]: "Multilayer Feedforward Networks Can Learn Arbitrary Mappings: Connectionist Nonparametric Regression with Automatic and Semi-Automatic Determination of Network Complexity," UCSD Department of Economics Discussion Paper.

White, H. [1989a]. *Estimation, Inference and Specification Analysis*. New York: Cambridge University Press, forthcoming.

White, H. [1989b]: "Some Asymptotic Results for Learning in Single Hidden Layer Feedforward Networks," *Journal of the American Statistical Association*, forthcoming.

White, H. [1989c]: "An Additional Hidden Unit Test for Neglected Nonlinearity in Multilayer Feedforward Networks," *Proceedings of the International Joint Conference on Neural Networks*, Washington, D.C. 1989. San Diego: SOS Printing, II:451-455.

White, H. and J. Wooldridge [1989]: "Some Results on Sieve Estimation with Dependent Observations," in W. Barnett, J. Powell and G. Tauchen, eds., *Nonparametric and Semiparametric Methods in Econometrics and Statistics*. New York: Cambridge University Press, forthcoming.

Wooldridge, J. [1989]: "Some Results on Specification Testing Against Nonparametric Alternatives," MIT Department of Economics Working Paper.

Statistical Neural Network Analysis Package (SNNAP)

Vincent L. Wiggins
RRC, Inc.
3833 Texas Avenue, Suite 285
Bryan, TX 77802
409/846-4713

Metrica, Inc.
3833 Texas Avenue, Suite 207
Bryan, TX 77802
409/846-4713

Jeff Grobman, Lt., USAF
Larry T. Looper
Human Resources Directorate
Manpower and Personnel Research Division
Brooks Air Force Base, TX 78235-5000
512/536-3648

Abstract — *Neural network techniques offer the ability to "discover" complex, interacting, and nonlinear relations from examples of system or individual behavior. The Armstrong Laboratory has developed an MS Windows based statistical neural network package (SNNAP) to ease the development of neural network models and implement results of prior research. SNNAP's network architectures are introduced and its training, analysis, and visualization facilities are demonstrated on an airman task performance example problem.*

INTRODUCTION

The field of neural networks encompasses a wide range of interdisciplinary topics that has recently experienced an explosion of both theoretical and applied research. While neural networks are often used for optimization problems, the current research emphasizes the ability of neural networks to extract features from examples of system or individual behavior. In this sense, the networks are used for problems typically approached with statistics, econometrics, clustering, and pattern recognition techniques. Common applications include system control, personnel or system flows, time-series projection, decision modeling, selection, identification, and fault detection.

The major advantage which neural networks bring to these problems is the ability to extract nonlinear relations and interactions among inputs without prior knowledge of specific functional forms. In fact, it can be shown that several neural network architectures can support the approximation of any continuous relationship (Hornik, Stinchcomebe, & White, 1989). This allows neural networks to be used as statistical models of complex behaviors where linear models are inappropriate or the form of the relationships is not known. As demonstrated in prior research (Wiggins, Engquist, and Looper, 1992a), this ability has allowed the networks to surpass the performance of some established personnel models developed with more traditional techniques.

Despite the obvious advantages, this highly flexible nature of neural networks can itself cause some problems. Neural network models are subject to over-fitting the data on which they are trained and this can produce models which perform very poorly out-of-sample. In addition, it can be difficult to understand or explain the nonlinear and interacting relations captured by a neural network model. Both of these issues are addressed in a software platform which has recently been developed by the Air Force Armstrong Laboratory

The Statistical Neural Network Analysis Package (SNNAP) provides a software environment for developing and analyzing neural network models of decisions, time-series phenomenon, system control, and other input-output relationships. It includes facilities to utilize three different network architectures, improve model selection, suggest network parameters, and visualize model response surfaces. SNNAP operates in the Windows 3.0 or 3.1 environment and makes extensive use of the Windows graphical interface. All software components were designed using object oriented techniques and the system is implemented in C++. This design allows new network architectures to be added to the system which will automatically take advantage of the existing visualization and other analysis facilities. The next two sections provide background on the network architectures and model selection methods implemented in SNNAP. This is followed by an example application where SNNAP's facilities are demonstrated on a model of airman performance. A more extensive coverage of SNNAP's facilities along with a more detailed treatment of the example application can be found in Wiggins, Borden, Engquist, and Looper (1992b).

NETWORK ARCHITECTURES

The heart of any neural network package is the network architectures which it supports. Neural networks are not a single technique, but a rapidly expanding field which has drawn from statistics, pattern recognition, neurobiology, statistical mechanics, and other fields. SNNAP implements three radically different network architectures, each of which has been successful in solving classification and continuous modeling problems. SNNAP allows several networks to be analyzed simultaneously. These networks can be selected to have similar architectures but different parameters or can be selected

from different architectures. More details can be found in the references and an overview of all three architectures is available in Wiggins, Looper, & Engquist (1991).

Back Propagation

Back propagation networks are the most widely and successfully applied network architecture. They have been employed in numerous areas and their performance has been compared to many traditional clustering, pattern matching, and statistical techniques. The success of back propagation in other areas of research and model building has recently been extended to personnel models (Wiggins et al. 1992a).

Back propagation networks utilize a layer of functions to develop relations between the inputs and outputs of a model. By using the output of some functions as inputs into other functions, complex functional forms can be generated. Typically these functions are arranged in layers, with the first layer receiving its inputs from the inputs to the model and each succeeding layer receiving inputs from the prior layer. This continues until the output layer is reached, and this layer produces the output (or outputs) of the model. When all connections between functions proceed from input to output, the network is referred to as a feed forward network. If connections are allowed back toward the inputs, the network is referred to as recurrent. A very simple example, using airmen reenlistment, is shown in Figure 1.

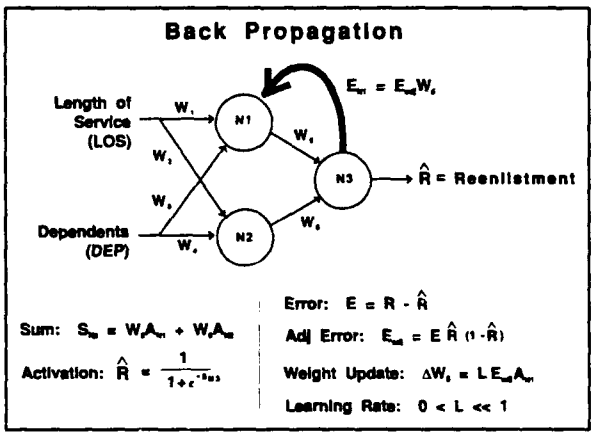


Figure 1. The back propagation method (reenlistment example). The neurons (N1, N2, and N3) are used to model reenlistment probability as a function of LOS and number of dependents.

The weights or function coefficients are designated by the W_i terms in the figures. Back propagation neurons are usually modeled as simple inner products between the inputs and the neuron weights with the result passed through a nonlinear transformation (or activation function). The most common activation transformation is the sigmoid or logistic curve (which is computed in Figure 1), although any monotonic function can be used. SNNAP provides a sigmoid, hyperbolic tangent,and linear activation functions. As a fourth activation function SNNAP includes product units which are particularly well suited to capturing interactions among model inputs (see Durbin & Rumelhart, 1989)

During the course of training, the weights in a network are changed to improve the ability of the network in predicting the observed outputs from the supplied inputs. The actual weight adjustment is made adaptively by successively presenting each training exemplar to the network and adjusting the weights slightly to improve performance on that single exemplar. A clever application of the chain rule of derivatives (see Rumelhart, Hinton, and Williams, 1986) allows the errors at the output layer to be propagated back to the hidden layers. The entire process proceeds to minimize the sum of squared errors using gradient descent over the entire network weight space. This adaptive process is performed many times for each observation in the training set and a single pass through the training data is termed an epoch. The rate at which the weights are adjusted is determined by two parameters which must be set by the researcher. SNNAP includes a module to "suggest" parameter settings for all network architectures.

SNNAP allows both recurrent and feed forward back propagation networks to be specified and trained. While feed forward networks are used for most applications, recurrent networks are particularly appropriate for time series data or other problems with a structure in time. The recurrent connections in the network allow the development of an internal structure relating current outputs to a representation incorporating both past and current inputs. The implementation of recurrent back propagation in SNNAP is a form of the simple recurrent network (SRN) developed by Elman (1990).

Probabilistic Neural Networks (PNNs)

A second major class of neural networks implemented in SNNAP are based on the estimation of probability density functions (PDFs) from the training data. These probabilistic neural networks (PNNs) are a direct neural representation of the statistically based Parzen windows (Parzen, 1962). They are typically applied to classification problems where one must identify a binary or categorical outcome (e.g. reenlist vs. separate vs. extend).

The PNN develops PDFs in the input space by placing a Gaussian kernel (other kernels are possible) over each observation in a data set. The kernels are then summed to produce a PDF for the class. This process can produce distributions of virtually any shape. The smoothness of

the distribution is determined by the assumed variance of the kernels placed over each observation. This variance is usually referred to as the smoothing factor for PNNs. The effect of different smoothing factors on a simple one-dimensional distribution can be seen in Figure 2. Each of the distributions shown in the figure were derived using different scaling factors from the same 5 data points. Computation of the PDFs is covered in detail in Specht (1990) and Wiggins et al. (1991).

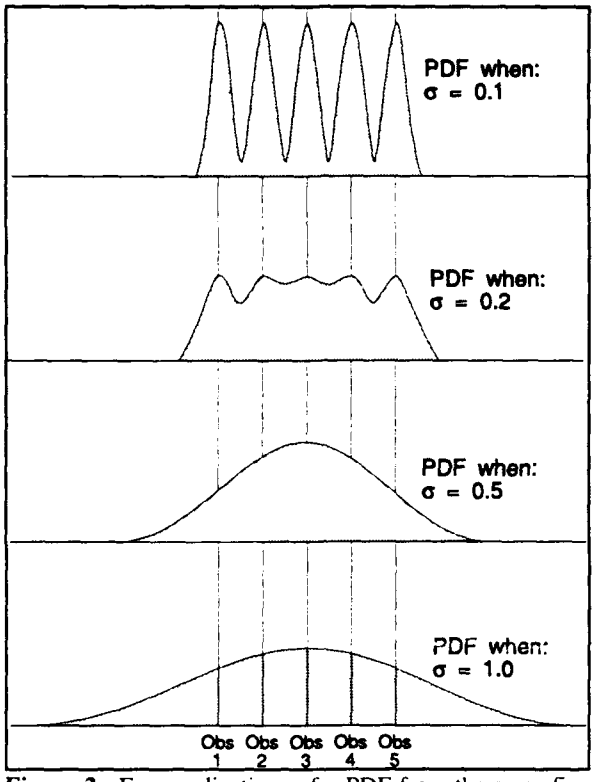


Figure 2. Four realizations of a PDF from the same five observations. Each realization uses a different smoothing factor.

Once a PDF has been generated, a new exemplar can be selected into one of the classes based on the relative heights of the class PDFs when evaluated at the input values for the new exemplar. The class with the highest density in the neighborhood of the exemplar is selected as the most likely class for the new exemplar. This process can also involve a priori weights applied to each of the classes. SNNAP supports this weighting and uses tne relative proportion of training exemplars in each class as the default a priori weights. SNNAP also extends the classification process to produce the probability (based on the PDFs) of a new exemplar falling into each of the possible classes.

SNNAP implements a third variant of PNNs which is used primarily to support analysis of the other networks. This network uses the PDF directly to estimate the relative density of data in any area of input space. This allows the analyst or researcher to determine if the estimation sample contains sufficient data in an area of the response surface which is of interest. If little training data exists in an area of input space, this reduces the confidence in the projected outcome.

Learning Vector Quantization (LVQ)

The learning vector quantization (LVQ) network was developed by Kohonen (1984) and is also a classification network. The network has been applied to several contrived problems and has often proven superior to standard classification techniques (Kohonen, Barna, & Chrisley, 1988). In several personnel areas, Wiggins et al. (1992a) found the LVQ to improve on the performance of regression and probit models but to perform somewhat worse than back propagation models. In general, the LVQ requires considerably less training time than back propagation and this may be a factor in some applications.

The LVQ network bears a strong resemblance to the K-means clustering algorithm (Duda & Hart, 1973), but has some features which improve its performance in classification tasks. The LVQ network operates by generating a set of reference vectors (or neurons) and placing them in the input space. These reference vectors are located at coordinates in the input space and serve as attractors for all exemplars which fall in their neighborhood. This can be seen in Figure 3, which shows a simple reenlistment model. In the top of the figure a hypothetical distribution of reenlisters and separators is shown. In the bottom of the figure, six reference vectors are placed in the two dimensional input space (3 to reenlistment and 3 to separation). Each reference vector has an area of influence within which all exemplars are assigned to the vector. A new exemplar to be projected is assigned to the nearest reference vector (usually computed by the Euclidian distance).

Training in an LVQ network involves determining the locations of the reference vectors in input space. If these locations were chosen to minimize within exemplar input variance and maximize between exemplar input variance, LVQ would exactly reproduce the K-means results. However, LVQ uses the actual classes of the training data exemplars to determine optimal class separation boundaries.

The primary parameter which must be designated with the LVQ architecture is the number of neurons or reference vectors. In general, this number can fluctuate over a fairly wide range and produce reasonable results. SNNAP's expert system is also configured to suggest a number of neurons given the problem type and number of training exemplars.

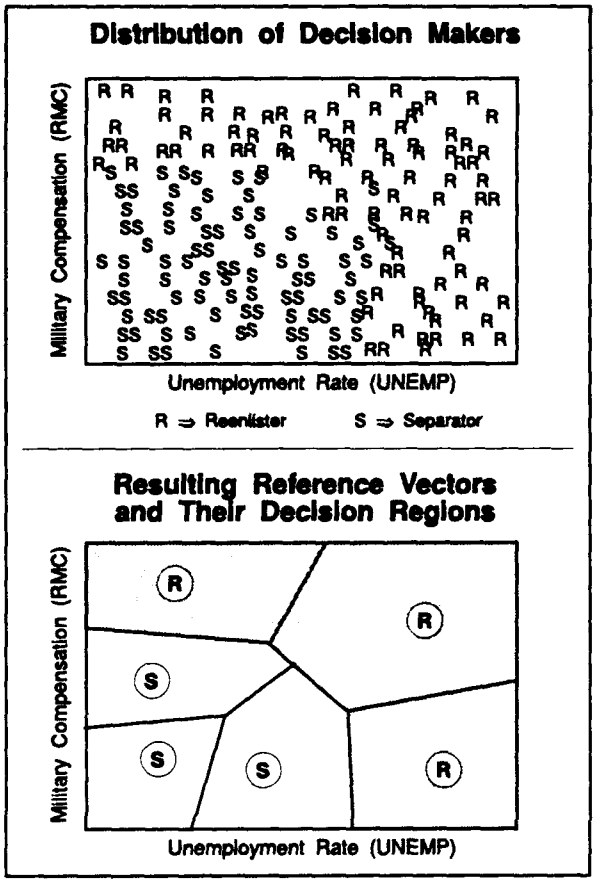


Figure 3. Decision boundaries formed by an LVQ network. Reenlistment decisions are modeled as determined by RMC and unemployment.

MODEL SELECTION

The ability of neural networks to produce complex and nonlinear relations between model inputs and outputs is one of their greatest assets. However, this ability can cause problems if the training data set contains a large stochastic component (i.e. the data has a large unexplained component or is noisy). When confronted with a noisy training data set, a neural network has the capability to "memorize" the noise in the data. Noisy training data leads to a problem similar to over-fitting with regression models containing high order terms. The network's performance may be very good in-sample (even flawless); however, when confronted with cases not in the training data, the network performs very poorly.

This ability to perform out-of-sample is referred to as the generalization problem. In all studies performed on personnel data, some method of preventing over-fit has been absolutely essential in developing models which generalize outside of the training sample (Wiggins et al., 1992a). The problem can be easily visualized using an example with back propagation training (see Figure 4). Back propagation is an adaptive process and requires many passes through a data set (epochs) for the network model to complete training. With slow training rates, performance always improves within the training sample. However, if performance is tracked on a hold-out or validation sample, this performance may degrade significantly beyond a certain point in training.

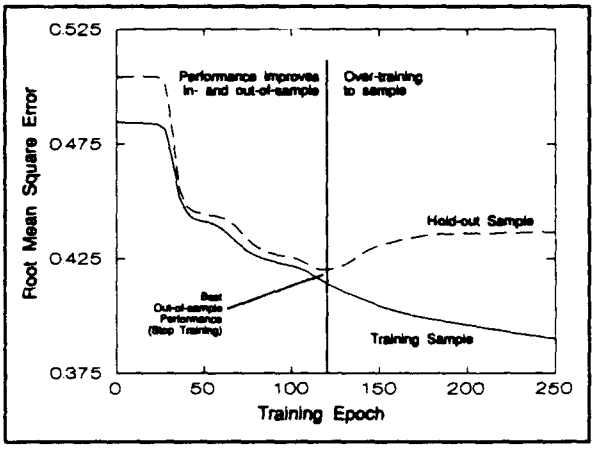


Figure 4. In-sample and out-of-sample training paths for back propagation training.

SNNAP provides facilities for saving a copy of a back propagation network each time a hold-out sample error basin (such as the one in Figure 4) is encountered during training. This is an extension to the early stopping training heuristics suggested by several researchers (Wiggins et al, 1992a; Morgan & Bourlard, 1990; Rumelhart, 1990). In the simple example shown in Figure 4, the hold-out sample performance (dashed line) has a single minimum point. In practice, several minimum "basins" can be encountered and the researcher would usually choose the one with the smallest root mean square error.

In addition to improving the predictive capability of networks, the performance on a validation sample provides some measure of confidence when interpreting the relations the network displays between model inputs and outputs. Standard statistics employed with regression models are not applicable to neural networks and the extremely flexible form of network architectures makes in-sample performance statistics meaningless. Hold-out or validation sample performance provides a quantitative measure of a network model's predictive ability.

AIRMAN PERFORMANCE EXAMPLE

An analysis of airman performance and its relation to aptitude and experience will be used to demonstrate SNNAP facilities and their application to a specific

problem. The approach will focus on using SNNAP to analyze the problem rather than theoretical, institutional, or data considerations. Several of the facilities available in SNNAP will be demonstrated with a more complete treatment available in Wiggins et. al. (1992b). All of the remaining figures are screen captures of actual SNNAP displays. For graphical output SNNAP also supports a **COPY** command to produce windows metafile (WMF) vector output for other windows applications. All of SNNAP's facilities and options are accessed using standard and custom Windows menu bars and dialog boxes. In general, these are very simple operations and only the results will be shown below.

Following the work of Lance, Hedge, & Alley (1987) and Vance, MacCallum, Coovert, & Hedge (1989) this example will be based on walk through performance test (WTPT) results. The WTPT is an objective measure of performance based on the ability to correctly complete critical steps in performing a specific task. At the Air Force Specialty (AFS) level, eight specialties were evaluated across several specific tasks with trained observers evaluating the performance of each step within each task.

This example will focus on a single task in AFS 324x0 (Precision Measuring Equipment Specialists). Specifically, hands-on performance on the task "Calibrates Distortion Analyzers" (designated H645) is analyzed. The proportion of task steps performed correctly is used as the performance metric (task H645 involves 30 steps); more details on the WTPT methodology can be found in Hedge (1984) and Hedge & Teachout (1986).

Table 1.
Variables in the Performance Model

Variables	Descriptions
H645per	Percent of steps completed correctly on the "Calibrates Distortion Analyzers" task (output/dependent variable).
Mp	Mechanical selector AI percentile
Ap	Administrative selector AI percentile
Gp	General selector AI percentile
Ep	Electronic selector AI percentile
h645num	Number of times the "Calibrates Distortion Analyzers" task was performed by the job incumbent prior to the WTPT.

As a measure of aptitude, all four of the Selector AI scores are used. These four scores are composites of the 10 Armed Services Aptitude Battery (ASVAB) sub-test scores. The number of times an airman had performed the "Calibrates Distortion Analyzers" task is used as a measure of task specific experience. This value was self-reported by the job incumbents when the WTPT was administered. All of the variables used in the analysis are summarized in Table 1. Complete information on these variables was available for 124 of the 140 airman administered the WTPT. The basis for model development will be these 124 observations with 1 output variable and 5 input variables.

Specifying the Variables

SNNAP currently accesses standard ASCII files as input. Standard Windows dialog boxes are used to select the name of these files and format files describing their contents. The user is allowed to choose the variables from the data set which are to serve as the inputs (independent variables) and the outputs (dependent variables) for the model. As shown in Figure 5, the selected input variables are those discussed earlier and documented in Table 1 (the data set contains additional variables which will be ignored). SNNAP top level menu bar is also visible in this figure. The output variable is the proportion of steps correctly completed in the hands-on portion of the "Calibrates Distortion Analyzers" task. Clicking on the **OK** button confirms the selected variables.

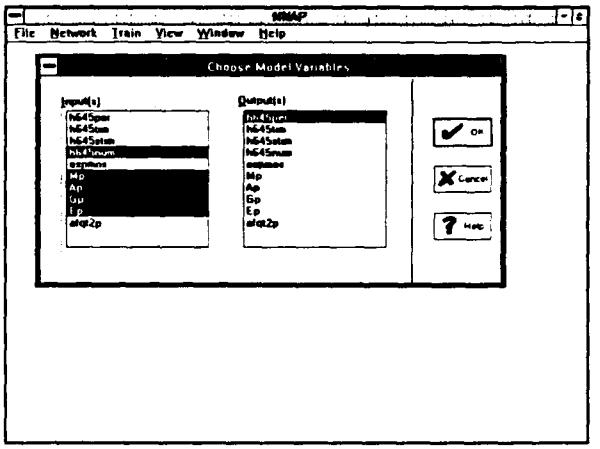


Figure 5. Selecting the input and output variables for a network.

After selecting the model's input and output variables, the user can specifying several types of sub-sampling to provide for on-line validation of the model's performance. In this case, a one-fifth random sample of the data set was designated as a hold-out sample in all of the analyses which follow.

Least Squares Baseline

Before proceeding to the development of a neural network model of task performance, an Ordinary Least Squares (OLS) model will be estimated to provide a baseline for the network model. Some form of benchmark model is extremely important in applying neural networks as they provide no intrinsic statistics on their own performance. Knowledge of the in- and out-of-sample performance of a baseline model can also help in assessing the progress of neural network training.

As seen in Figure 6, OLS appears as a "network type" when selecting the type of model to be built. SNNAP's modular architecture is designed to ease the addition of new network types or other analysis techniques such as logit, discriminant analysis. Even such nonlinear statistical techniques as classification and regression trees (CART) or projection pursuit could be added to the system. Once added, these techniques would automatically have complete access to SNNAP's analysis and visualization facilities.

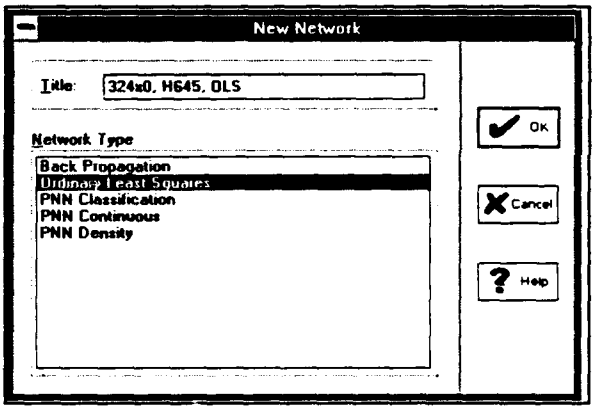
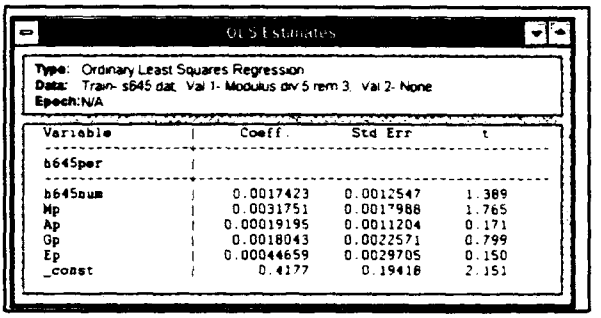


Figure 6. Selecting the Ordinary Least Squares "network" type".

The results of the OLS analysis can be seen in Figure 7. As can be seen in the figure, only mechanical aptitude (Mp) is statistically significant at the 5% level, with task experience (h645num) just significant at the 10% level. However, the overall relationship between aptitude, experience, and job performance is rather tenuous in the OLS model.

In all cases, the OLS facility excludes the validation sample (or samples) from the estimation process. This behavior will be exploited later to compare OLS and neural network model performance. It should be noted that the OLS model need not use the same variables as the neural network models. In particular, many existing regression models apply logs, squares, or other transformations to their input terms (or output). While it is uncommon to apply such transformations to neural network inputs, separate variables containing the transformed variables can be included only in the OLS models. This makes it possible to compare neural network performance against many existing models completely within the SNNAP environment.



Variable	Coeff.	Std Err	t
b645per			
b645num	0.0017423	0.0012547	1.389
Mp	0.0031751	0.0017988	1.765
Ap	0.00019195	0.0011204	0.171
Gp	0.0018043	0.0022571	0.799
Ep	0.00044659	0.0029705	0.150
_const	0.4177	0.19418	2.151

Figure 7 OLS job performance model results.

A Back Propagation Model

With a baseline model in hand, we can proceed in developing a neural network model of task performance. For this example, the back propagation architecture will be used. To date, this architecture has consistently shown the good performance in personnel research (Wiggins et al., 1992).

To a point, the back propagation model is specified in precisely the same manner as the OLS model just developed. However, a special dialog box is used to specify options which are specific to the back propagation architecture. These options will not be discussed here but include such items as the number of hidden layers, type of transfer functions, structure of layer connections, training rate, momentum factor, and input or output scaling. Some of these options can be seen in the dialog box shown in Figure 8.

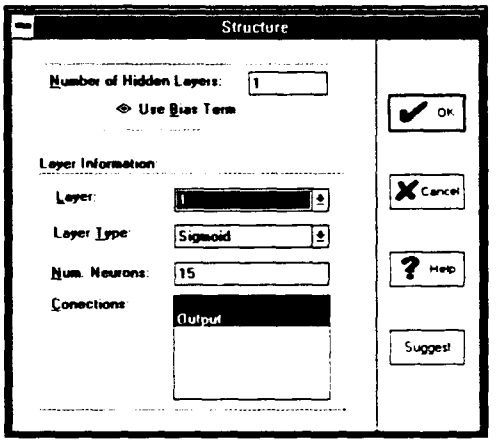


Figure 8. Specifying the structure of a back propagation network.

For this analysis, instead of "hand specifying" the network's parameters, the **Suggest** button will be used to invoke SNNAP's expert system. SNNAP then queries the user on one or two topics to insure that its preliminary analysis of the data is correct. Using this information, the expert systems generates a default architecture which can be accepted as is or modified by the user. In this case the suggested structure and parameters will be used in the following analysis. Used in this manner, SNNAP makes developing neural network models almost as effortless as developing linear regression models.

As discussed earlier, back propagation networks are trained to exemplars (or observations) by making multiple passes (epochs) through a data set while making small adjustments to the network's weights. SNNAP automatically tracks this training process by reporting on the root mean square error (RMSE) for both the training and hold-out sample. Figure 9 shows the training path for the airman performance data with the bottom line showing the RMSE for the training sample and the top line the RMSE for the hold-out (or validation) sample

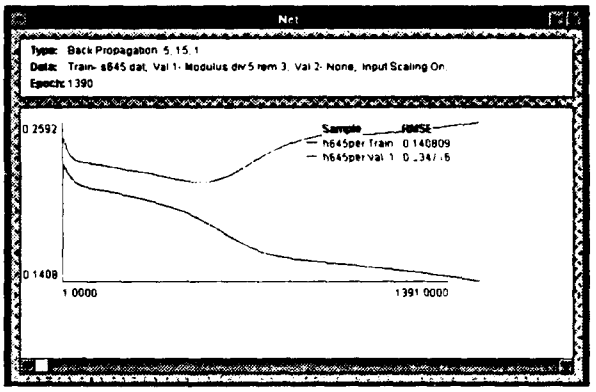


Figure 9. Training and validation (hold-out) sample performance paths during back propagation training.

The validation sample RMSE displays the characteristic shape for out-of-sample performance with noisy data. At first, the hold-out sample improvements in performance almost parallel the performance on the training data. However, the performance improvements eventually flatten and RMSE actually begins to rise. This upward sloping portion of the validation sample training path can be construed as over-training or over-fitting the training sample data. If the goal of the model is to extract underlying features from the data or to project job performance for individual not in the sample, over-training will be contrary to this goal. SNNAP contains facilities to automatically save the state of the network at points during training which are likely to produce the best generalization (out-of-sample performance). One of these points is the minimum of validation sample RMSE. An interface is provided to restore the network's state to any of these training points. All of the analysis and visualization facilities can be used on any of the saved network states, or even on the current state of a network during training.

Comparing Model Performance

The first analysis we will perform involves comparing the training and validation sample performance of the OLS and back propagation (BP) network models. Training and validation sample performance statistics are available from the main menu bar for any network or regression model. Figure 10 displays the results of computing summary statistics on the OLS and BP network airman performance models just developed (statistics for the OLS model appear above the BP statistics).

OLS Statistics h645per

Statistic	Training	Validation 1
RMSE	0.1987	0.2132
Actual Mean	0.8902	0.8853
Network Mean	0.8902	0.8686
Actual Std. Dev	0.2105	0.2279
Network Std. Dev.	0.0692	0.0766
TIC	0.1555	0.1687
TICB	0.0000	0.0061
TICV	0.5050	0.5036
TICC	0.4950	0.4903
R squared	0.1082	0.1247
Janus Quotient	0.9444	0.9356
Correlation	0.3289	0.3616

Net: h645per

Statistic	Training	Validation 1
RMSE	0.1757	0.1992
Actual Mean	0.8902	0.8853
Network Mean	0.8894	0.8577
Actual Std. Dev	0.2105	0.2279
Network Std. Dev.	0.1030	0.1533
TIC	0.1372	0.1577
TICB	0.0000	0.0192
TICV	0.3743	0.1400
TICC	0.6257	0.8408
R squared	0.3033	0.2358
Janus Quotient	0.8347	0.8742
Correlation	0.5545	0.5226

Figure 10. Performance summary statistics for the OLS and BP models of airman performance.

Outside of the means and standard deviations (which are informative but do not compare performance), all of the statistics shown in Figure 10 are derived from or related to the sum of squared prediction errors. Each of the RMSE, TIC, R-squared, Janus Quotient, and Correlation are different scaled measures of the error. The Janus Quotient and TIC represent perfect prediction with

zero and larger values represent worse performance (TIC limited by infinity, the Janus Quotient by 1). The R-squared and actual/predicted correlation assume the value 1 for models which predict perfectly.

As can be seen in the figure, the BP network fits the actual task performance measure better both in the training and validation samples. The differences can be seen most plainly in the R-squared and the correlation coefficient where the scale of these measures improves their resolution in the error range of these models. It is interesting to note that the 0.3616 correlation for the OLS model on the 25 validation sample observations represents an insignificant correlation at the 5% level. Alternately, the .5226 validation sample correlation for the BP model is significant at the 5% level. Likewise the validation sample R^2 for the BP model is almost twice the R^2 for the OLS model (the in-sample R^2 is almost triple the OLS model). Overall, the BP model appears to be capturing significantly more structure than the OLS model and this structure improves its out-of-sample performance.

Comparing the actual and network model standard deviations, it can be seen that the OLS model shows much less variability in its predictions than exist in the actual data. While still smaller than the actual standard deviation in the H645per variable, the network produces considerably more variation in its response than the OLS model. The importance of this can be seen by examining the TICV or variance component of the TIC. For the OLS model, about 50% of the prediction error, as measured by the TIC can be attributed to lack of variation in the OLS predictions. Alternately, only 35% training sample and 14% validation sample TIC error is attributed to lack of variation in the BP model. The BP model comes much closer to reproducing the variability of the performance variable.

Viewing the Response Surface

Having established that the BP model is capturing features in the data which allow it to perform considerably better than a linear regression, it now becomes interesting to investigate the structure of the BP model. This can be done using the visualization facilities in SNNAP. These facilities are accessed very simply using standard dialog boxes and the "point-and-click" windows user interface. Several options are available on each view (such as log scales and selecting viewing regions) but only the results will be explored here.

One of the more interesting aspects of the current models is the simple experience/performance profile or time to proficiency relationship. A graph of job performance vs. task experience will illustrate this on the job training aspect of task performance. A view of this relationship for individuals with typical scores on all selector AIs is shown in Figure 11.

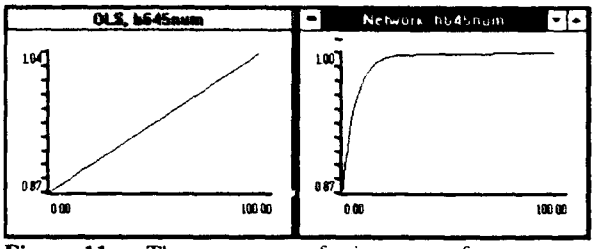


Figure 11. The response of airmen performance to different levels of task experience. OLS model on the left, back propagation model on the right.

The two models clearly have a different opinion of the impact of task experience on task performance. While both models agree that the proportion of steps correctly completed is 0.87 for those with no task experience and about 1.00 for those with 100 repetitions performing the task, they differ radically in how the 100% performance or proficiency is obtained. The BP network model postulates that proficiency on the task improves dramatically early in the experience path with complete proficiency obtained with fewer than 20 repetitions. Alternately, the OLS model, restricted by its linear form, postulates a steady improvement over the entire experience path. It should be noted that the form suggested by the network is not well approximated by simple transformations such as logs. It is most similar to a functional form requiring nonlinear estimation techniques and which is notoriously unstable to estimate.

OLS

h645num	h645per
0.000	0.875
10.000	0.891
20.000	0.907
30.000	0.923
40.000	0.938
50.000	0.954
60.000	0.970
70.000	0.986
80.000	1.002
90.000	1.018
100.000	1.033

Network

h645num	h645per
0.000	0.867
10.000	0.976
20.000	0.995
30.000	0.998
40.000	0.999
50.000	0.999
60.000	0.999
70.000	1.000
80.000	1.000
90.000	1.000
100.000	1.000

Figure 12. Tabular view of task performance over a range of task experience levels. OLS model on the left, BP network model on the right.

The differences in the experience/performance training paths can be seen numerically by toggling the views from Figure 11 to tables (see Figure 12). The tables show the

modeled level of performance for various numbers of task experience repetitions prior to testing. Again, both methods model very similar levels of performance for those with no task experience (0.872 for OLS and 0.874 for BP). However, they model decidedly different pathways to full proficiency. At just over 5 repetitions, the network model projects almost 95% of steps completed correctly. The OLS model projects over 42 repetitions required to reach this same performance.

When looking at Figure 11, one should keep in mind that the graphs shown are merely a 2-dimensional slice out of a 6-dimensional response surface. For the OLS model, this point is irrelevant. The slope of the line shown will be the same regardless of the value of the other 4 variables (Mp, Ap, Gp, and Ep). Of course, as the other 4 variables change; the level, or intercept, of the line will of course vary according to the positive or negative coefficients on the other 4 variables. The interpretation of the graph produced by the BP network is radically different. The trained network model may contain features which cause not just the level, but also the impact of h645num to change as the other variables change. For example, the shape of the network curve in Figure 11 may be different for high aptitude airmen and low aptitude airmen.

One way of directly visualizing the interactions just discussed is to examine 3-dimensional slices of the model's response surface. To do this, two input variables are selected instead of the single (task experience) variable from the views shown above. For example, if both task experience (h645num) and mechanical aptitude (Mp) are selected as inputs and views are produced for both the OLS and BP models, the results are as shown in Figure 13.

The graph of the OLS model is the expected plane in 3-D space. However, the BP network model shows a much more interesting structure. Those with very high mechanical percentile scores require almost no task experience to perform the "Calibrates Distortion Analyzers" task perfectly. Those with very low mechanical aptitude require many repetitions to achieve perfect performance (this is a task with a very high performance rating across individuals). It can also be seen that performance improves dramatically with very few repetitions for those with low and middle Mp percentile scores. While all mechanical percentile groups eventually produce maximum performance (as measured here), the amount of task training required to attain this performance is directly related to aptitude as measured by mechanical percentile. The BP network also shows a much wider performance response over the same aptitude and experience input values: BP model (.58 to 1.00), OLS model (.72 to 1.08). This is consistent with the higher variation seen in the BP model statistics.

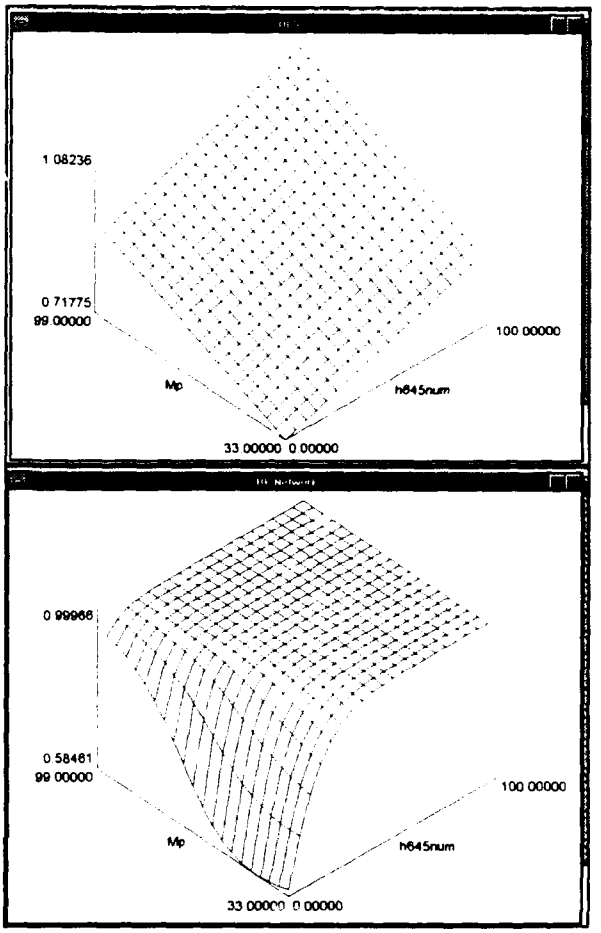


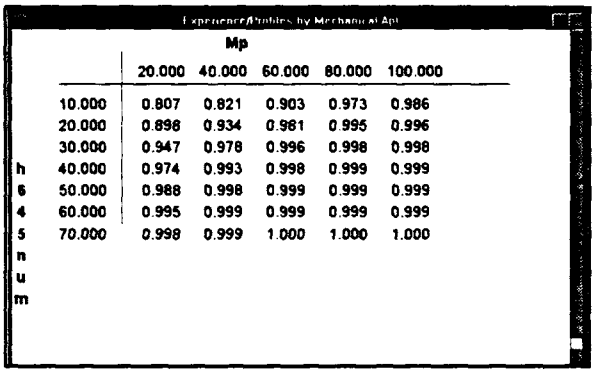
Figure 13. The response of airman performance to a range of levels of task experience (h645num) and mechanical aptitude (Mp). OLS model on top and back propagation on the bottom.

As with the 2-dimensional views, 3-dimensional views can be toggled to tables. When the BP network view from Figure 13 is toggled, the table shown in Figure 14 results. This table numerically shows the effect of task experience (h645num) and mechanical aptitude (Mp) on task performance. Each column of the table represents an experience/performance profile evaluated at mechanical aptitude percentiles of 20, 40, 60, 80, and 100 respectively. The table provides precise numerical verification of the analysis of the surface plot. In many instances, the results of the table can be used directly in other software (such as airman selection or allocation packages) which requires tabular scales as input.

Getting a Different Perspective

SNNAP offers many options for helping to interpret and analyze the three dimensional graphical views of network response. These options include rotations, scaling, height shading by color, and shading by slope.

Figure 15 demonstrates an option which is particularly useful in interpreting the BP network response surface from Figure 13. In this case, only the lines in the Y-axis direction (task experience) are displayed. Each line now represents a specific mechanical percentile score. In effect, several of the 2-D graphs shown in the right half of Figure 11 have been superimposed on the same graph. The only difference between each line is the Mp score



h645num	Mp 20.000	40.000	60.000	80.000	100.000
10.000	0.807	0.821	0.903	0.973	0.986
20.000	0.898	0.934	0.961	0.995	0.996
30.000	0.947	0.978	0.996	0.998	0.998
40.000	0.974	0.993	0.998	0.999	0.999
50.000	0.988	0.998	0.999	0.999	0.999
60.000	0.995	0.999	0.999	0.999	0.999
70.000	0.998	0.999	1.000	1.000	1.000

Figure 14. Tabular view of BP Network response surface relating task experience (h645num) and mechanical aptitude (Mp) to task performance.

This graph makes very apparent, the different task experience-performance profiles of airmen with different Mp scores. Those with lower scores have heavily curved lines which begin at just under 60% of steps correctly completed and rise rapidly to 100% of steps completed. Those airmen with high Mp scores begin their jobs with nearly complete proficiency. Several of the other options available can be seen in the superimposed dialog box in the figure.

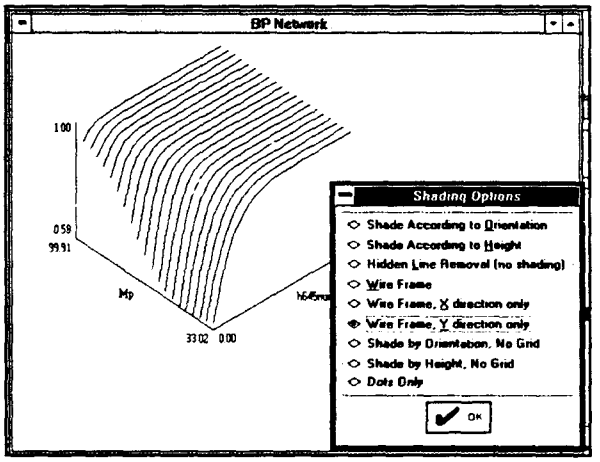


Figure 15. Y-axis wire frame view of BP model. Response of airman performance to task experience and mechanical aptitude.

Automatic Surface Scanning

SNNAP contains facilities to automatically search a response surface and note any distinctive features in the surface. It searches for linear, log-linear, linear-log, and log-log responses over the entire area for which data is available. Any of these functional relations which remain constant over the range of the scan can be identified. Any other relation is flagged as unidentified. The scan also searches for interactions among inputs where the impact of one input on an output depends on the level of another input. As with all other facilities, surface scanning is initiated from the main menu bar. Any feature documented by the surface scan can be quickly realized as a 3-dimensional surface view just by selecting the feature and the **View** option.

Figure 16 shows the entire SNNAP work surface with the results of scanning the BP network model in the window on the left. The icons at the bottom of the screen represent several of the analyses and windows which have been shown in the figures above. They can be recalled simply by selecting their respective icons.

The highlighted line in the scan result window indicates that there is an interaction between mechanical aptitude (Mp) and administrative aptitude (Ap) in determining job performance. A graph of this interaction has been generated by selecting the **View** button on the scan window. An option has been used to darken the surface of the graph where the slope is largest to emphasize the interaction between the two selector AI aptitude measures. As can be seen, lower mechanical aptitude individuals compensate with higher administrative aptitude to reach the high performance plateau (the light surface at the top of the graph). However, the curved surfaces indicate that the two aptitude measures do not contribute equally and are not additive in determining performance.

Multi-line Views

Another analysis component of SNNAP provides facilities for analyzing specific cohorts or even individuals using the completed models. Figure 17 demonstrates this facility using the BP network model of task performance. Three groups have been defined: 1) individuals scoring 40 on all selector AI percentiles, 2) individuals scoring 70 on all selector AI percentiles, and 3) individuals scoring 95 on all selector AI percentiles. (These are arbitrary groupings and the SNNAP user is free to define as many groups as desired.) Given the defined groups, graphs can be made relating any input to the model's output (or outputs). In Figure 17, experience/performance profiles are displayed for each of the three cohorts.

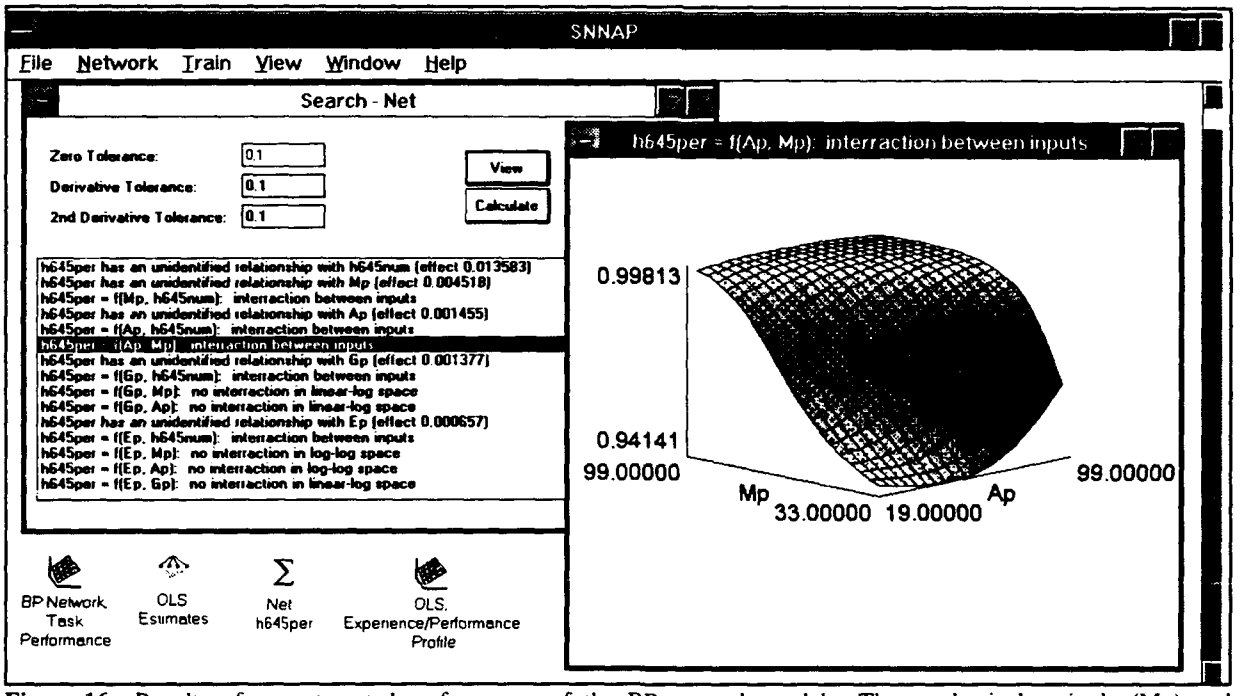


Figure 16 Results of an automated surface scan of the BP network model. The mechanical aptitude (Mp) and administrative aptitude (Ap) interaction has been graphed.

As seen earlier, aptitude has a significant impact on initial task performance and the path to task proficiency. The form of the experience/performance paths could play an important role in both the selection and technical training processes. In particular, this relationship would be valuable in determining the optimal level of task training during technical training. As with all other views, tables of the graphed values can be obtained.

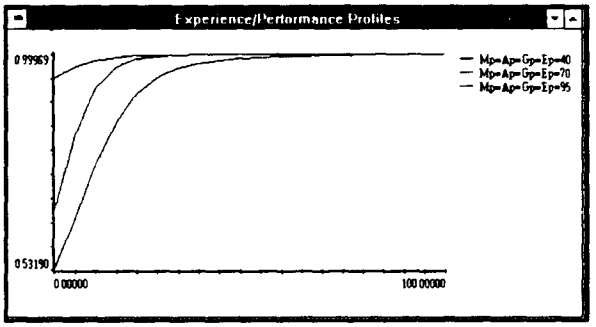


Figure 17. Experience/Performance profiles for three groups: all selector AIs equal 40, all selector AIs equal 70, and all selector AIs equal 95.

ANALYSIS SUMMARY

In this example, the ability of the network model to project the performance of airmen not in the training sample was superior to the ability of the regression model. On the basis of this performance, an analysis of the network model's response surface revealed several interesting features.

While this analysis was limited to a single task in one AFS, many of the model's features would have significant policy implications if they were applied to selection and training. The Mp score appears to be a better indicator of task performance than the selector AI for the career field (Ep). All aptitude groups are capable of excellent task performance if task specific experience is sufficient. This hands-on training is not nearly as important for high Mp aptitude airmen as it is for those with lower Mp aptitude. In particular, hands-on training for the "Calibrates Distortion Analyzers" task is particularly effective for low and middle Mp aptitude airmen.

CONCLUSION

SNNAP is an environment for designing, training, and analyzing neural networks. It provides extensive facilities for visualizing and quantifying the relationships captured in a trained neural network. The performance of network models can be examined both in- and out-of-sample; and this performance can be compared to regression models within the SNNAP environment. SNNAP also implements automated facilities for suggesting network design and

analyzing the surface of trained networks. It incorporates training heuristics to improve the ability of the network models to generalize to exemplars data outside the training data.

As demonstrated in the example problem and prior research (Wiggins et al., 1992), neural networks can reveal complex nonlinear structure in models of many personnel decisions, behaviors, and systems. This structure often offers deeper insight into relationships and interactions among model determinants. As seen in the task performance example and prior research on reenlistment rates the nonlinear features developed by networks often have significant implications for policy decisions. SNNAP provides the capability to easily search for and illustrate these nonlinear features. The software provides an integrated environment to exploit the capabilities of neural networks in areas where model generalization and a deep understanding of the modeled relations is required.

REFERENCES

Duda, R, & P. Hart, *Pattern Classification and Scene Analysis*, John Wiley and Sons, New York, 1973.

Durbin, R. & Rumelhart, D.E. (1989). Product units: a computationally powerful and biologically plausible extension to backpropagation networks. *Neural Computation*, **1**, 1, 133-142.

Elman, J.L. (1990). Finding structure in time. *Cognitive Science*, **14**, 179-211.

Hedge, J.W. (1984). *The methodology of walk-through performance testing*. Paper presented at the annual meeting of the American Psychological Association, Toronto.

Hedge, J.W., & Teachout, M.S. (1986). *Job performance measurement: a systematic program of research and development* (AFHRL-TP-86-37, AD-A174 175). Brooks AFB, TX: Training Systems Division, Air Force Human Resources Laboratory.

Hornik, K., Stinchcomebe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), 359-366.

Kohonen, T. (1984). *Self-organization and associative memory* (3rd ed.). New York: Springer-Verlag.

Kohonen, T., Barna, G., & Chrisley, R. (1988). Statistical pattern recognition with neural networks. *IEEE International Conference on Neural Networks, San Diego, California, July, 1988*, **1**, I - 61-88.

Lance C.E., Hedge, J.W., & Alley, W.E. (1987). *Ability, experience, and task difficulty predictors of task performance* (AFHRL-TP-87-14). Brooks AFB, TX: Manpower and Personnel Division, Air Force Human Resources Laboratory.

Morgan, N, & Bourlard, H. (1990). Generalization and parameter estimation in feedforward nets: some experiments. *Neural Information Processing Systems 2*, Touretzky, D.S. (ed.), San Mateo, CA: Morgan Kaufmann Publishers, 630-637.

Parzen, E., "On Estimation of a Probability Density Function and Mode", *Annals of Mathematical Statistics*, vol. 33, pp. 1065-76, 1962.

Rumelhart, D.E. (1990). Brain style computation: neural networks and connectionist AI (oral presentation). Las Vegas TIMS/ORSA Joint National Meeting, May 7-9, 1990.

Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986). Learning internal representations by error propagation, in *Parallel distributed processing: explorations in the microstructure of cognition*, D.E. Rumelhart & J.L. Mclelland (Eds.). Cambridge, MA: MIT Press, 213-362.

Specht, D.F. (1990). Probabilistic neural networks. *Neural Networks*, 3(1), 109-118.

Vance, R.J., MacCallum, R.C., Coovert, M.D., & Hedge, J.W. (1989). Construct models of task performance. *Journal of Applied Psychology*, **74**, 3, 447-455.

Wiggins, V.L., Looper, L.T., & Engquist, S.E. (1991). *Neural networks and their application to air force personnel modeling* (AL-TR-1991-0031). Brooks AFB, TX: Human Resources Directorate, Manpower and Personnel Division, Armstrong Laboratory.

Wiggins, V.L., Engquist, S.E., & Looper, L.T. (1992a). *Applying neural networks to Air Force personnel analysis* (AL-TR-1991-0118). Brooks AFB, TX: Human Resources Directorate, Manpower and Personnel Division, Armstrong Laboratory.

Wiggins, V.L., Borden, K.M., Engquist, S.E (1992b). *Statistical neural network analysis package (SNNAP): overview and demonstration of facilities* (AL-TP-1992-0055). Brooks AFB, TX: Human Resources Directorate, Manpower and Personnel Division, Armstrong Laboratory.

Personnel Analysis Applications of Neural Networks

Vincent L. Wiggins
RRC, Inc.
3833 Texas Avenue, Suite 285
Bryan, TX 77802
409/846-4713

Metrica, Inc.
3833 Texas Avenue, Suite 207
Bryan, TX 77802
409/846-4713

Larry T. Looper
Human Resources Directorate
Manpower and Personnel Research Division
Brooks Air Force Base, TX 78235-5000
512/536-3648

Abstract — *Neural network techniques offer the ability to "discover" complex, interacting, and nonlinear relations from examples of system or individual behavior. When compared with some current statistical models of reenlistment and other decision behavior, the networks were found to provide substantially better predictive performance. The reenlistment response surfaces of these neural network models were found to agree with risk and uncertainty theories.*

INTRODUCTION

Personnel researchers have applied many modeling and analytic techniques to quantify the decisions, behaviors, and flows observed in personnel systems. In recent years artificial neural network (ANN) techniques have demonstrated some impressive results in modeling other complex systems and in classification tasks (see Caudil, 1990). The success of ANNs in these areas and their potential for application to personnel modeling lies principally in their ability to automatically detect nonlinear and interacting relations among the inputs and output(s) of a system or observed behavior. Most personnel models require the determination of a relation between a set of inputs (known characteristics or conditions) and a target variable such as a decision, capability, flow, or stock. Traditional analytic techniques require that the form of this relation be specified by an analyst before the empirical estimation of the relationship. ANNs allow more complex relations to be developed directly from observed behaviors of the system or group of individuals under analysis.

AIRMAN REENLISTMENT

The first personnel area examined is the reenlistment decision of first-term airmen. Specifically, given an airman eligible to make a reenlistment decision, the airman's demographic characteristics, Air Force policy, and economic conditions at the time of the decision; what is the likelihood the airman will reenlist? A model capturing this type of decision process serves as the cornerstone of most personnel inventory models. In addition, this area serves as a very good test-bed for the capability of ANNs. As reenlistment has historically been of critical planning importance to the Air Force, it has engendered much research activity: Saving et al.(1985); Kohler (1988), and others.

While the reenlistment decision has been heavily researched, virtually all of the models tested have been linear in their input terms. Many researchers have employed logit or probit analysis which imposes a fixed nonlinearity on the output, but still has no inherent flexibility. It was hoped that the flexible form of the ANN models would capture a more complex mapping from the known characteristics of the airman and the decision environment onto the reenlist/separate decision.

Model and Data

The reenlistment model chosen is taken from the research of Stone et al. (1990). This model is particularly appropriate for ANN analysis because it retains as inputs the separate components of the pecuniary factors: military compensation (RMC), selective reenlistment bonus (SRB), and civilian wages. The commonly used Average Cost of Leaving (ACOL) construct, which aggregates all pecuniary factors into a single ACOL term (Warner & Goldberg, 1983), would prevent an ANN from searching for potentially more fruitful combinations of pecuniary factors.

Stone et al. estimated their model over the January 1975 through March 1982 time period and validated the resulting equations over the April 1982 through March 1986 time period. Each of the major Air Force Specialties (AFSs) were modeled using a separate probit equation estimated on individual level data for all airmen in an AFS eligible to make a decision during the estimation sample time frame. The resulting probit equations were used to predict the reenlistment decisions of airmen eligible to make decisions over the validation sample time frame. The Stone model employs 18 independent variables to capture the economic and policy conditions at the time each airman made a reenlistment decision. These variables included pecuniary factors (discounted RMC, civilian wages, SRB, and employment rates), demographic factors (race, dependents, marital status, and gender), aptitude, experience, and quarter in which the decision was made.

These variables reflect a mature reenlistment model with long-term refinement of the model through two previous revisions. In this sense, it should provide a stringent benchmark against which ANNs can be compared. The data used in developing the ANN models is the same data used by Stone with the current analysis restricted to three 5-digit career fields and two 2-digit career fields as seen in Table 1.

Table 1. AFS Codes and Specialties

AFS Code	Description
272x0	Air Traffic Control
316x1	Missile System Maintenance
426x2	Jet Engine Mechanic
30xxx	Communications-Electronics Systems
47xxx	Vehicle Maintenance

Modeling Methods

Three ANN architectures were compared against the probit results: back propagation (BP; Rumelhart, Hinton, and Williams, 1986), probabilistic neural network (PNN; Specht, 1990), and learning vector quantization (LVQ; Kohonen, 1984). The specifics of these ANN architectures are discussed in detail in the respective references. In general, the PNN and LVQ networks utilize local information and smoothing to generate the response surface of a model. The importance of BP in our results and the heuristics required to obtain good performance with the architecture necessitates some explanation.

BP networks utilize layers of simple nonlinear functions to construct complex functional relations. Despite the fact that these simple functions typically employ the same nonlinearity (sigmoidal), it can be shown that the layered architecture is capable of producing any continuous nonlinear function (Hornik, Stinchcomebe, & White, 1989). As shown in Equation 1, the first layer of functions receive their inputs directly from the model's independent variables. The functions in the ensuing network layers receive their inputs (X_i) from the outputs (P_i) of other functions.

$$P_i = \frac{1}{1 + e^{-CX_i}} \qquad (1)$$

Where:

C is the vector of coefficients or weights.

X_i is the vector of inputs or independent variables for observation i.

The weights or coefficients in a BP network are determined using a supervised learning procedure in which the network adapts to the inputs and desired outputs by error correction. The most common error metric involves minimizing the sum of squared prediction errors over the training exemplars. Rumelhart was among several researchers who independently developed a direct gradient method of propagating an error measure back though a layered network to adjust the function coefficients.

The freedom of a BP model to fit the inputs to the desired output is related to the number of processing elements it employs and the number of layers into which they are organized. Typically the complexity of a BP solution is constrained by limiting the number of processing elements in the network to enhance the generalization capability (or out-of-sample performance) of a network. Given the large stochastic component in most personnel data sets, it is important to limit the complexity of the trained network model. Without some constraint, it is quite likely that a BP network will simply "memorize" the exemplars without formulating a model which performs well on individuals or exemplars with new combinations of characteristics.

An alternative to limiting the number of processing elements, is limiting the amount of training time allowed. The BP method is adaptive and requires many (often thousands) of passes though a data set (epochs) before training is complete. Several researchers (Morgan & Bourlard, 1990) have suggested stopping the training early as a means of improving out-of-sample generalization.

An example of over-training on actual reenlistment data can be seen in Figure 1. Training past the vertical line in the Figure causes the out-of-sample performance to degrade — the root mean square error (RMSE) increases. This portion of the training path could be categorized as memorizing the noise in the training sample rather than extracting relevant features. By observing the network's performance on a hold-out sample, on which training is not performed, the training process can be terminated before memorization begins.

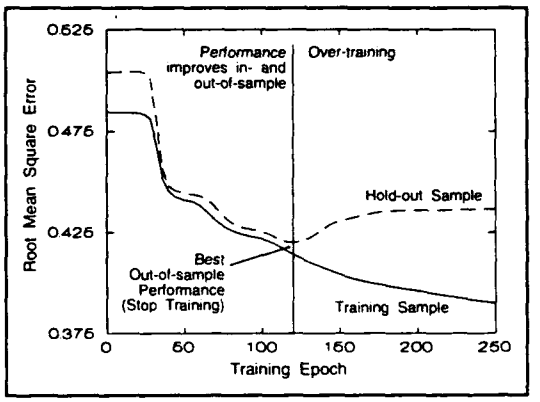


Figure 1. Training path for back propagation. Training sample (solid line) and hold-out sample (dashed line) performance as the number of training passes through the training data set increases.

For personnel research, this early stopping process has been found to be much more successful than limiting the number of processing elements. The heuristics for applying early stopping in our research are outlined in Table 2; more computational intensive re-sampling methods should be able to further improve generalization.

Table 2. Back Propagation Training Stopping Methods

Method	Description
BP Hold	Compute the validation sample RMSE after each training pass through the estimation sample. Choose the amount of training which produces the smallest RMSE on the validation sample. This is a best case method which cannot be obtained in practice when the validation sample is unknown at the time the model is developed.
BP Tri-sample	1. Randomly split the original estimation sample into pre-estimation and pre-validation samples. 2. Train only on the pre-estimation sample while tracking the RMSE on the pre-validation and pre-estimation samples. 3. Save the pre-estimation RMSE at the training point where the pre-validation RMSE is best. 4. Re-train the network on the full estimation sample (both the pre-estimation and pre-validation samples). Stop training when the RMSE from the full estimation sample matches the one saved in Step 3.
BP Temporal	Same as BP Tri-sample except the step 1 split into pre-estimation and pre-validation samples is such that these two samples cover separate time periods.

Reenlistment Results

Following the work of Stone et al., the validation sample was taken to be airmen eligible to make a first-term reenlistment decision between the dates April, 1982 and March 1986. Airmen making a decision between January 1975 through March 1982 were used to estimate (or train) the probit and ANN models. In each case, the models resulting from estimation or training on the estimation sample were used to produce predictions of the decisions of those airman in the validation sample.

The simulation R^2 was employed to measure the performance of each model's predictions. An R^2 of one implies perfect fit and zero implies a model which performs no better than the in-sample mean.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (Predicted_i - Actual_i)^2}{\sum_{i=1}^{n} (ActualMean - Actual_i)^2} \qquad (2)$$

The out-of-sample (validation sample) results of the probit models are compared with the BP models using the three training stopping heuristics in Table 3. In virtually all cases, the BP models performed substantially better than the probit models currently in use. When BP was able to track performance on the validation sample (BP Hold), it produced the best projections. However, the temporal sub-sampling method (BP Temporal) produced comparable results on all AFSs except 316x1 (and did not require information on the validation sample). The results when tracking a random estimation sub-sample (BP Tri-sample) were more mixed, but still considerably better than the probit models for all specialties except 426x2. For the AFSs analyzed, the BP Temporal method explained 35 to 100% more out-of-sample variation than the probit models. Results for the PNN and LVQ architectures are fell between the probit and BP results in all specialties and are not reported.

Table 3. Validation Sample Results (April 1982 through March 1986)

AFS	Simulation R^2 by Modeling Technique			
	Probit	BP Hold	BP Tri-Sample	BP Temporal
272x0	.139	.222	.154	.205
316x1	-.194	.116	-.173	-.035
426x2	.269	.368	.141	.365
30xxx	.155	.244	.241	.316
47xxx	.198	.331	.300	.312

AGGREGATE ACCESSION AND RETENTION

A second reenlistment area examined involves aggregate retention rates modeled simultaneously with accession rates in a time-series model of personnel flows. On an aggregate level, the Air Force personnel system has three major flow rates: non-prior service accessions (NPS), prior service accessions (PS), and separations. In the current model, only voluntary separations are modeled using the reenlistment rates for first-term (RELRT1) and second-term (RELRT2) airmen. As with prior research on individual reenlistment, prior aggregate flow models (Ash, Udis, & McNown, 1983; DeVany & Saving, 1982; and Stone, Saving, Turner, & Looper 1991) employed regression techniques and structural relations which could be made linear in the regression inputs.

Time-series Model and Data

An aggregate model including the four flow rates described above (NPS, PS, RELRT1, and RELRT2) served as the basis for developing ANN models. This model was taken from the Stone et al. (1991) model which was extensively tested over out-of-sample time periods and proved far superior to the rather poor accession results

obtained by Ash et al. (1983). The model is structural in the sense that each dependent variable has an equation with a specified form and set of independent variables. Each equation included explanatory terms for relative military to civilian wages, and unemployment levels. Other factors such as production recruiters, DEP waiting time, and force level and accession goals were included in the accession equations. Eligibility and early out factors were included in the reenlistment equations. The prior researchers employed ordinary least squares (OLS) to separately estimate each flow rate equation and generalized least squares (GLS) to simultaneously estimate the four equations.

The Stone group estimated the equations using monthly data over one time period, October 1979 through September 1987, and validated their performance over two time periods, January 1979 through October 1979 and October 1987 through September 1988 (FY 88). Only the performance on the latter validation sample is examined here, using the earlier validation sample to determine when training should be stopped.

A training heuristic similar to that used on individual reenlistment was applied to the aggregate rate models. Again, the BP Hold method stopped training when performance was best on the actual validation sample (FY 1988). Using the BP Temporal method, training was terminated when performance was best on the other temporal hold-out sample (January 1979 to September 1979). A third training heuristic takes advantage of the empirical observation that most out-of-sample performance optimums occur at a particular point during in-sample training (when the second derivative of the in-sample RMSE with respect to the amount of training switches from negative to positive). The second occurrence of such an inflection point during training is designated as the BP Inflection network. This method utilizes no information from outside the training sample.

Table 4. Validation Sample Performance (FY 1988)

Modeling Technique	Simulation R^2			
	NPS Access Rate	PS Access Rate	1st-term Reenlist Rate	2nd-term Reenlist Rate
OLS	.618	.378	.288	.569
GLS	.606	.317	.237	.323
BP Temporal	.487	.633	.683	.736
BP Hold	.647	.633	.774	.736
BP Inflection	.644	.550	.772	.436

Time-Series Results

A comparison of the out-of-sample performance of the two regression techniques and three variations on BP are presented in Table 4.

The improvement of the ANN techniques over the regression methods was quite marked. In several cases, ANN models explained more than twice the out-of-sample variations when compared with the OLS or GLS models. Two of the three BP methods also performed slightly better on the NPS accession rate. Although not typically as strong as the other two BP training methods, BP Inflection outperformed the regression techniques in all cases except OLS on second-term reenlistment.

Figure 2 displays the FY 88 out-of-sample projections of OLS and BP Inflection. While both project well, the OLS projection misses the upswing in reenlistment by a month, the downturn by 2 months, and projects rates in excess of 100% for two months. The ANN projection captures both the onset and downturn in the reenlistment rate quite accurately.

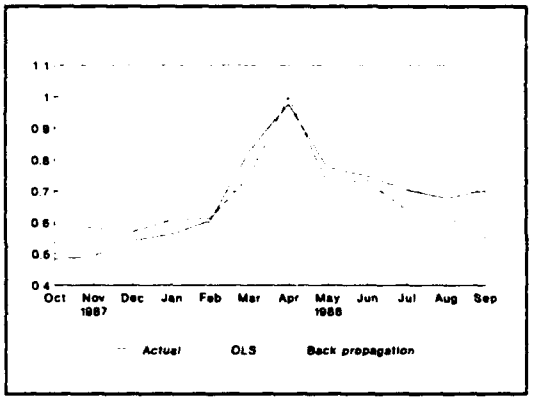


Figure 2. Actual and out-of-sample projections of first-term reenlistment rates for October 1987 through September 1988, OLS and BP (inflection) models.

ANN Response Surfaces

Given the ability demonstrated by BP networks in out-of-sample projections, it is interesting to analyze the factors which set the networks apart from the regression techniques. In particular, the networks must be capable of capturing relationships between the independent variables and aggregate rates not specified in the regression models. Two of the principal inputs in each rate equation are a measure of the civilian employment level and relative military to civilian wage. The impacts of employment and relative wages on each of the aggregate rates, as modeled by the ANNs, are presented in Figures 3 through 6.

Figure 3 displays two nonlinear but essentially non-interacting impacts on first-term reenlistment. Looking strictly along the unemployment axis, there are two relatively flat surfaces where changes in unemployment have little effect on the reenlistment rate — below 6%

unemployment and above 8.5% unemployment. Increases in unemployment above 8.5% do not substantially affect reenlistment; likewise, decreases below 6% have almost no impact.

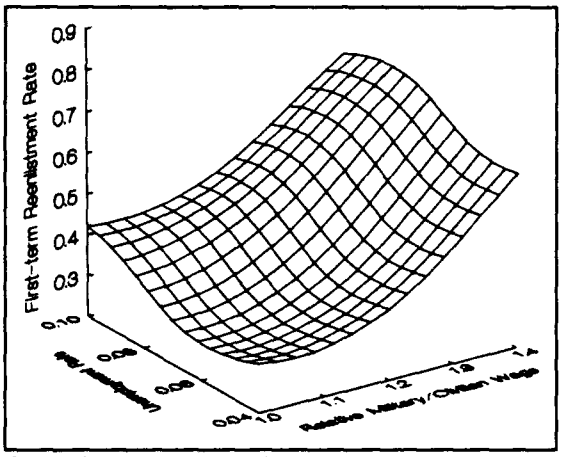


Figure 3. Response of first-term reenlistment rate to unemployment levels and relative military to civilian wage, estimated by the BP Inflect ANN model.

The ANN modeled relation between relative wages and first-term reenlistment is also nonlinear but of a different form. When military compensation exceeds the civilian wage by less than 10%, changes which keep the relative wage below that level have virtually no effect. As relative wages move above 1.1 the effect of a given change in relative wage produces significant changes in the reenlistment rate. The form of these nonlinearities would have a dramatic impact on the implication of a change in Air Force compensation policy or shifting economic conditions.

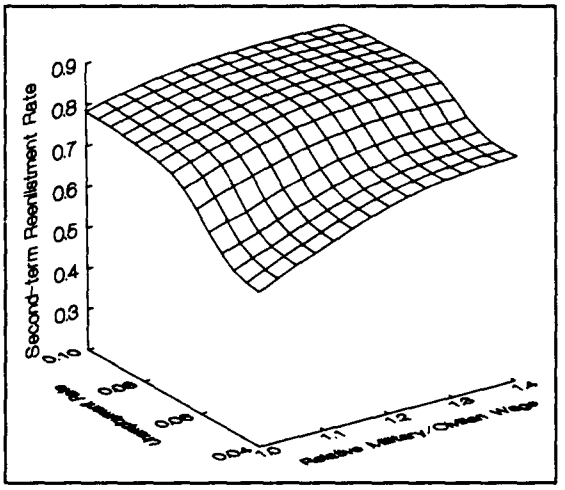


Figure 4. Response of second-term reenlistment rate to unemployment levels and relative military to civilian wage, estimated by the BP Temporal model.

With second-term reenlistment (Fig. 4), a soft threshold phenomenon is again seen relating reenlistment and unemployment. Below 5% and especially above 7.5% unemployment, changes in the unemployment rate have minimal effect on second-term reenlistment. For second-term reenlistment, the effective range has shifted down 1% from the transition range observed for first-term reenlistment. This shift would reflect an increased risk-aversion exhibited by the older group. As expected, the reenlistment rate for second-term decision makers is consistently high and relatively unaffected by changes in military compensation.

The NPS accession rates shown in Figure 5 display two linear, non-interacting but important impacts from the two variables. This result is to be expected given the relative performance of the ANN and regressions models. Of the four modeled rates, the out-of-sample results were most similar for NPS accessions. Essentially, the ANN has reinforced the original modeler's implicit assumption that no nonlinear features were present in the NPS accessions model.

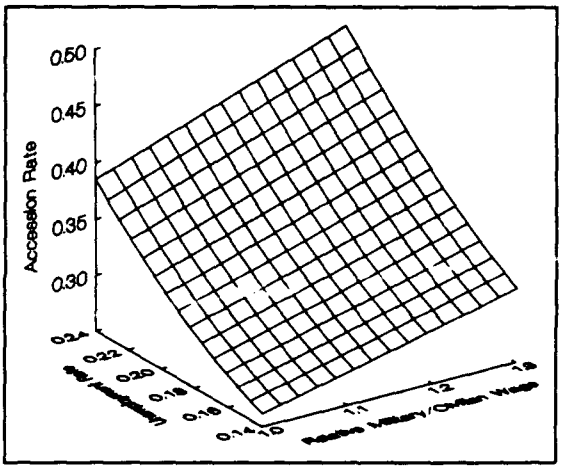


Figure 5. Response of the NPS accession rate to unemployment levels and relative military to civilian wage, estimated by the BP Inflect ANN.

Prior services accession rates (Fig. 6) demonstrate considerable interaction between unemployment rate and relative wage. The unemployment level has a dramatic impact on how potential PS accessions respond to changes in relative military to civilian wages. When unemployment is very low, changes in military compensation have little effect until the military wage exceeds its civilian counterpart by over 20%. However, with high unemployment, the impact of military compensation begins before the relative difference is 10%. When unemployment is high, the impact of changing military compensation is much larger and increases faster. This is precisely the type of behavior one would expect from a labor group already entrenched in the work-force. High relative wages and changes in those relative wages have much less effect on those who already hold jobs.

Most of these features were poorly approximated by the constant effects constraint of linear models or the

constant elasticity of log-log models. Although the network was relatively unconstrained in its ability to fit the training data, the features developed were well behaved and extrapolate smoothly. In each case, the nonlinear and interacting features "postulated" by the network model were extremely plausible and often more intuitively appealing than constant or constant elasticity effects over the entire range of an input variable.

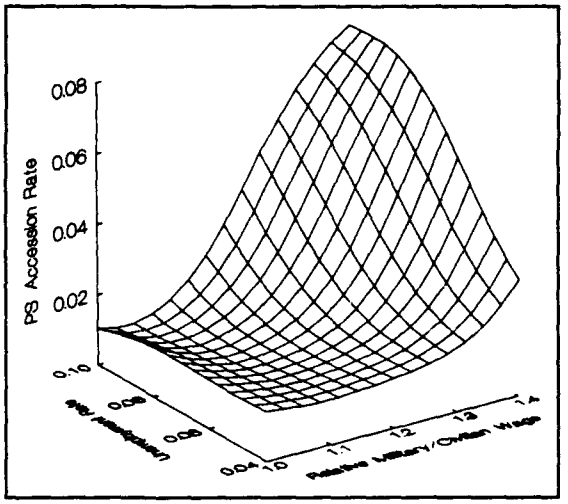


Figure 6. Response of the PS accession rate to unemployment levels and relative military to civilian wage, estimated by the BP Inflect ANN.

A common complaint among researchers modeling time series data involves changes in model structure. When an equation is estimated over one time period, its coefficients may substantially differ from those obtained over a different time period. A "change in structure" is usually blamed for these differences. However, a glance at Figure 6 will show that a linear model estimated over a time period of high unemployment would produce a substantially different result than one estimated over a period of moderate unemployment. While this is typically considered a change in structure over time and is the bane of effective projection, the ANN model suggests an alternate interpretation. The model structure has remained constant; it merely contains a richer, more nonlinear structure, than the original estimator was capable of capturing. When networks can capture some of this richer structure, they can be expected to perform significantly better than regression techniques.

CONCLUSIONS

Overall, ANNs have demonstrated the ability to significantly improve on the performance of some existing models. This ability is directly related to the amount of nonlinear or complex structure in the system being estimated. In lieu of the constant impact or constant elasticity of most regression methods, a successfully trained network offers more insight into the structure of the problem (as seen in Figures 3 - 6). With the proper tools, the interrelations and features developed by a network can be made available as a more realistic model of the process being analyzed.

A critical concern to any research on personnel or other highly stochastic systems involves methods to prevent over-fitting. The heuristics employed in this research were critical to ANN performance and proved successful at stopping training before the network's ability to generalize outside the estimation sample declined. Prevention of over-fitting is an area which has received limited attention in the literature and many refinements are possible. In spite of the extremely successful results obtained in some areas of this study, care must be taken when applying ANNs. Comparisons should always be made with more traditional techniques and out-of-sample testing performed to ensure the ANN has not obtained a degenerate response surface.

REFERENCES

Ash, C., Udis, B., & McNown, R.F., (1983). Enlistments in the all-volunteer force: A military personnel supply model and its forecasts. *American Economic Review*, 73(1), 145-155.

Caudil, M. (ed.) (1990). *International Joint Conference on Neural Networks, January 15-19, 1990, Washington D.C., IEEE & INNS*

DeVany, A.S., & Saving, T.R., (1982). Life-cycle job choice and the demand and supply of entry level jobs: some evidence from the Air Force. *The Review of Economics and Statistics*, LXIV(3), 457-465

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), 359-366.

Kohler, D.F., (1988). *Using survivor functions to estimate occupation-specific bonus effect.* (R-3348-FMP). Santa Monica, CA: The RAND Corporation.

Kohonen, T. (1984). *Self-organization and associative memory* (3rd ed.). New York: Springer-Verlag

Morgan, N. & Bourlard, H. (1990). Generalization and parameter estimation in feedforward nets: some experiments. *Neural Information Processing Systems* 2, Touretzky, D.S. (ed.), San Mateo, CA: Morgan Kaufmann Publishers, 630-637.

Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986). Learning representations by back-propagating errors. *Nature*, **323**, 533-536.

Saving, T.R., Stone, B.S., Looper, L.T., & Taylor, J.T., (1985). *Retention of Air Force enlisted personnel: An empirical examination.* (AFHRL-TP-85-6). Brooks AFB, TX: Manpower and Personnel Division, Air Force Human Resources Laboratory.

Specht, D.F. (1990). Probabilistic neural networks. *Neural Networks*, 3(1), 109-118.

Stone, B.S., Looper, L.T., & McGarrity, J.P., (1990). *Validation and reestimation of an Air Force reenlistment analysis model.* (AFHRL-TP-89-55). Brooks AFB, TX: Manpower and Personnel Division, Air Force Human Resources Laboratory.

Stone, B.S., Saving, T.R., Turner, K.L., & Looper, L.T. (1991). *Integrated economic and behavioral modeling of accession and retention* (AL Technical Paper in review). Brooks AFB, TX: Human Resources Directorate, Manpower and Personnel Division, Armstrong Laboratory.

Warner, J.T., & Goldberg, M.S., (1983). The influence of non-pecuniary factors on labor supply: The case of Navy enlisted personnel. *The Review of Economics and Statistics*, XX(Y), 26-35.

A Comparison of Ordinary Least-Squares - Linear Regression and Artificial Neural Network - Back Propagation Models for Personnel Selection Decisions *

by

W. A. Sands
Director, Personnel Systems Department
Navy Personnel Research and Development Center
San Diego, CA 92152

and

C. A. Wilkins
Department of Psychology
Baylor University
Waco, TX 76798

Background

Dichotomous criteria are frequently important in the area of military personnel selection. Sometimes, even though the criterion of interest is available in continuous form, it is dichotomized for the sake of convenience. An important example of such a criterion is successful completion of an enlistee's first tour of obligated service vs. premature attrition. The efficacy of prediction models for forecasting these dichotomous criteria is a major technical issue in personnel selection.

Purpose

The purpose of the study described herein was to compare two alternative approaches to the problem of predicting a dichotomous criterion. The first approach involved an Ordinary Least-Squares - Linear Regression (OLS-LIN) model. The second approach involved an Artificial Neural Network - Back Propagation (ANN-BKP) model. A major objective of the study was to conduct it in a way that would enable the results to be generalized to a wide variety of personnel selection situations.

Approach

The authors decided to use computer-simulated data, rather than empirical data. Use of computer-simulated data had the advantage of

---

* The opinions expressed in this paper are those of the authors, are not official, and do not necessarily represent those of the Navy Department or Baylor University.

allowing relatively precise control over the bivariate distributions (i.e., the datasets were relatively "well-behaved" in a statistical sense). Use of any specific empirical dataset(s) would run the risk of involving some idiosyncratic aspects of a particular dataset, and could have seriously limited the ability to generalize the results.

The dimensions of the personnel selection situation included in this study were as follows.

1. Underlying functional form of the relationship between the predictor and criterion variables (linear vs. curvilinear).

2. Sample size.

3. "Error," or degree to which the individual data points deviated from the ideal functional form. This was measured by the standard deviation of the observations around the line portraying the underlying functional form. In the linear case, this "error" can be transformed to a validity coefficient; i.e., the correlation between the predictor and the criterion.

4. Base rate - the proportion of persons considered successful, prior to introducing a new selection instrument.

5. Selection ratio - the proportion of applicants selected for acceptance, based upon scores on the new selection instrument.

6. Sample split - the proportional allocation of persons in a total sample into two subsets: (a) the developmental sample and (b) the evaluation sample. Each model was developed on the former type sample and evaluated on the latter type sample.

The degree of error introduced into the distributions was chosen so as to produce the following validity coefficents in the linear case: .05, .25, .50, .75, and .90. Errors corresponding to these target validities were used to simulate total bivariate data distributions for three sample sizes: 100, 500, and 5000. This was done for each of the two functional forms: linear and curvilinear. Then, each total sample was divided into two subsets (development and evaluation), according to the following allocations: 20%-80%, 50%-50%, and 60%-40%.

At this point, there was a bivariate distribution of two continuous variables for each sample size, for each functional form. The simulated subjects were rank-ordered on the continuously distributed criterion variable. Then the continuous criterion was converted into a dichotomous variable representing success vs. failure, using the base rate under consideration. This procedure was followed separately for each base rate considered: .05, .25, .50, and .95.

An Ordinary Least-Squares - Linear Regression (OLS-LIN) model was determined for each development sample. These OLS-LIN models were used to predict criterion scores for each simulated subject in the complementary evaluation sample. In a similar fashion, an Artificial Neural Network - Back Propagation (ANN-BKP) model was trained on the simulated subjects in a development sample, then tested in the complementary evaluation sample. The ANN-BKP models developed employed the standard back propagation learning algorithm, with the following architecture: one input node, one hidden layer containing three hidden nodes, and one output node. The stopping criterion for training the ANN-BKP models on the development samples was 100,000 iterations.

A criterion score was estimated for each subject in the evaluation sample. This predicted criterion score was estimated using the development sample model. The subjects in the evaluation sample were rank-ordered by their estimated criterion score. Alternative selection ratios were applied, dividing the group into selectees and rejectees. The selection ratios employed were: .05, .25, .75, and .90.

At this point, the actual status of each subject was known for the predictor side (selection vs. rejection) and the criterion side (success vs. failure), for each combination of dimensions studied. This two-by-two situation produced four decision-outcome combinations: (1) correct acceptances - persons selected who subsequently succeeded, (2) erroneous acceptances - persons accepted who subsequently failed, (3) correct rejections - persons rejected who would have failed if they had been accepted, and (4) erroneous rejections - persons who would have succeeded if they had been accepted. This two-by-two table information was combined into the total number of correct decisions (correct acceptances and correct rejections) and the total number of erroneous decisions (erroneous acceptances and erroneous rejections). Finally, the proportion of correct decisions ("hit rate") was determined for each evaluation sample.

The hit rate in the evaluation sample was used as the measure of effectiveness for comparing the OLS-LIN models against the ANN-BKP models, under each combination of conditions (functional form, sample size, degree of error from the functional form, base rate, selection ratio, and development-evaluation sample split). Ordinarily, the statistical procedure for evaluating these comparisons would have been the appropriate t-test. However, the data in this study did not meet the assumptions required for this parametric test. Therefore, nonparametric tests were employed to assess statistical significance. Specifically, the McNemar test was used when the number of subjects in an evaluation sample classified differently by the OLS-LIN and ANN-BKP models was greater than, or equal to, ten. The binomial test was used when the number of subjects classified differently by the two procedures was less than ten.

Results

The two approaches (OLS-LIN and ANN-BKP) exhibited statistically significant differences ($p<.001$) in 62 comparisons. All of these significant differences occurred in the curvilinear cases; no significant differences were obtained for the two approaches when the underlying functional form was linear. Sixty-one of the 62 significant differences favored the ANN-BKP model over the OLS-LIN model. Fifty-six of the 62 significant differences were observed in the largest sample size situation (N=5000), 6 in the next sample size (N=500), and no significant differences were observed between the models when the sample size was 100. Significant differences between the models did not appear related to the other dimensions studied: base rates, selection ratios, and sample splits.

Discussion

A major advantage of the ANN-BKP approach is that the researcher does not need to know the underlying functional form involved in a prediction problem. In theory, the ANN-BKP model will discover the nature of the underlying functional form. The OLS-LIN model will perform quite well in a situation where the underlying relationship between the predictor and criterion is linear, but substantially less well when the underlying functional form is curvilinear. In actual research, the nature of the underlying functional relationships between variables is frequently unknown. The availability of an analytic tool that does not require that knowledge about a particular dataset would be highly advantageous.

Conclusion

The results of this study are very encouraging. The ANN-BKP models used in this study employed a single architecture, fixed values for the model parameters, and a single stopping criterion for training. In a real-life situation, each of these features would be varied to identify the best settings for the specific problem being addressed. Despite this lack of fine-tuning, the ANN-BKP models performed as well as the OLS-LIN models when the underlying functional form was linear. This is actually somewhat remarkable since the OLS-LIN is designed for the linear case, and the ANN-BKP models had to discover the linear relationship. In the curvilinear case, the ANN-BKP models outperformed the OLS-LIN models in 61 of the 62 cases where there was a statistically significant difference ($p<.001$) between the models. The ANN-BKP model appears to be a very powerful prediction tool that merits further research.

Linear and Neural Network Models for Predicting Human Signal Detection Performance from Event-Related Potentials: A Comparison of the Wavelet Transform with other Feature Extraction Methods

Leonard J. Trejo
Navy Personnel Research and Development Center
San Diego, CA 92152-7250

Mark J. Shensa
Naval Command Control and Ocean Surveillance Center, RDT&E Division
San Diego, CA 92152-5000

ABSTRACT

This report describes the development and evaluation of mathematical models for predicting human performance from discrete wavelet transforms (DWT) of event-related potentials (ERP) elicited by task-relevant stimuli. The DWT was compared to principal components analysis (PCA) for representation of ERPs in linear regression and neural network models developed to predict a composite measure of human signal detection performance. Linear regression models based on coefficients of the decimated DWT predicted signal detection performance with half as many free parameters as comparable models based on PCA scores and were relatively more resistant to model degradation due to over-fitting.

Feed-forward neural networks were trained using the backpropagation algorithm to predict signal detection performance based on raw ERPs, PCA scores, or high-power coefficients of the DWT. Neural networks based on high-power DWT coefficients trained with fewer iterations, generalized to new data better, and were more resistant to over-fitting than networks based on raw ERPs. Networks based on PCA scores did not generalize to new data as well as either the DWT network or the raw ERP network.

The results show that wavelet expansions represent the ERP efficiently and extract behaviorally important features for use in linear regression or neural network models of human performance. The efficiency of the DWT is discussed in terms of its decorrelation and energy compaction properties. In addition, the DWT models provided evidence that a pattern of low-frequency activity (1 to 3.5 Hz) occurring at specific times and scalp locations is a reliable correlate of human signal detection performance.

INTRODUCTION

Studies have shown that linear regression models may significantly explain and predict human performance from measures of ERPs elicited by stimuli presented in the context of a task (Trejo, Lewis, & Kramer, 1991; Trejo & Kramer, 1992). These models have used, as predictors, measures such as the amplitude and latency of ERP components (e.g., N1, P300). Other studies have used more comprehensive measures such as factors derived from principal components analysis and discriminant functions (Humphrey, Sirevaag, Kramer, & Mecklinger, 1990). Such models work best when they have been adapted to the individual subject, taking into account the temporal and topographic uniqueness of the ERP. Even then, the models often suffer from a limited ability to generalize to new data. In addition, the cost of developing and adapting such models for individuals is high, requiring many hours of expert analysis and interpretation of ERP waveforms.

Neural-network models for ERPs may be an improvement over linear regression models (DasGupta, Hohenberger, Trejo, & Mazzara, 1990; Ryan-Jones & Lewis, 1991). However, when neural network models have been based on traditional ERP measures, such as the sampled ERP time points or the amplitude of ERP components, the improvement in correlation between ERP measures and human performance has been small, typically about ten percent (Venturini, Lytton, & Sejnowski, 1992). Transformations of ERPs prior to neural network analysis, such as the fast Fourier transform (FFT), may improve neural network models (DasGupta, Hohenberger, Trejo, & Kaylani, 1990).

However, the FFT is not ideally suited for representing transient signals; it is more appropriate for narrow-band signals, such as sine waves.

The wavelet transform is well-suited for analysis of transients with time-varying spectra (Tuteur, 1989; Daubechies, 1990, 1992) such as the ERP. Discrete wavelet transforms (DWT) represent signals as temporally ordered coefficients in different scales of a time-frequency plane. More precisely, the DWT represents signals in a *time-scale* plane, where scale is related to — but not identical with — frequency. Scales are implemented by dilating a "mother wavelet" in the time domain. Each dilation is a doubling of the wavelet length in the time domain which results in a halving of the bandwidth in the frequency domain.

Thus a scale of the transform corresponds to one octave of signal bandwidth beginning with the smallest scale, i.e., the scale that corresponds to the highest frequencies represented in the signal. This scale, which is referred to here as *scale 0*, contains frequencies ranging from the Nyquist frequency (half the sampling rate) to one half the Nyquist frequency. As scales increase, the bandwidth decreases by a factor of two. For example, the bandwidth of scale 1 extends from 1/2 Nyquist to 1/4 Nyquist, and so on. The result of this successive halving of scale bandwidth is increasing frequency resolution (narrower bands) at larger scales (lower frequencies).

Because large scales represent low frequencies, fewer coefficients are required to represent the signal at large scales than at small scales. In fact, since the bandwidth decreases by a factor of two with each scale increase, the sampling rate or number of coefficients can also be halved with each scale increase. This process, called *decimation*, leads to an economical but complete representation of the signal in the time-scale plane. However, in some cases decimation may be undesirable, for example, when the temporal detail in a particular scale is of interest. In such cases, the undecimated wavelet transform may be computed (Shensa, 1992).

It is convenient to refer to the bandwidths of the scales in units of Hz, and this familiar unit will be used to make the following illustration. For a one-second long EEG signal with a bandwidth of 32 Hz and 64 time points, the first and smallest scale of the DWT would represent frequencies in the range from 16 to 32 Hz with 32 coefficients. The next larger scale would represent frequencies of 8 to 16 Hz with 16 coefficients. Successively larger scales would have the bandwidths and numbers of coefficients: 4-8 Hz/8, 2-4 Hz/4, 1-2 Hz/2, 0-1 Hz/1. A single additional coefficient would represent the DC level, for a total of 64 coefficients.

As with the discrete Fourier transform, with appropriate filters the DWT is invertible, allowing for exact reconstruction of the original signal. An important feature of the DWT, however, is that the coefficients at any scale are a series that measures energy within the bandwidth of that scale as a function of time. For this reason it may be of interest to study signals within the DWT representation and use the DWT coefficients of brain signals directly in modeling cognitive or behavioral data.

In this study, the effect of representing ERPs using the DWT was compared with more traditional representations such as raw ERPs, peak and latency measures, and factors derived using principal components analysis (PCA). The comparisons determined whether the DWT can efficiently extract valid features of ERPs for use in linear regression models of human signal detection performance. In addition, neural network models were tested to determine whether the relative efficiency and validity of the DWT and other ERP representations would be maintained with a non-linear method.The signal detection task was chosen because ERP-performance relationships in this task have been described and analyzed with linear regression models based on peak and latency measures of ERP components (Trejo et al., 1991; Trejo & Kramer, 1992).

METHOD

In an earlier study (Trejo et al., 1991), ERPs were acquired in a signal detection task from eight male Navy technicians experienced in the operation of display systems. Each technician was trained to a stable level of performance and tested in multiple blocks of 50-72 trials each on two separate days. Blocks were separated by 1-minute rest intervals. About 1000 trials were performed by each subject. Inter-trial intervals were of random duration with a mean of 3 s and a range of 2.5-3.5 s. The entire experiment was computer-controlled and performed with a 19-inch color CRT display.

Triangular symbols subtending 42 minutes of arc and of three different luminance contrasts (.17, .43, or .53) were presented parafoveally at a constant eccentricity of 2 degrees visual angle. One symbol was designated as the target, the other as the non-target. On some blocks, targets contained a central dot whereas the non-targets did not. However,

the association of symbols to targets was alternated between blocks to prevent the development of automatic processing. A single symbol was presented per trial, at a randomly selected position on a 2-degree annulus. Fixation was monitored with an infrared eyetracking device. Subjects were required to classify the symbols as targets or non-targets using button presses and then to indicate their subjective confidence on a 3-point scale using a 3-button mouse. Performance was measured as a linear composite of speed, accuracy, and confidence. A single measure, PF1, was derived using factor analysis of the performance data for all subjects, and validated within subjects. PF1 varied continuously, being high for fast, accurate, and confident responses and low for slow, inaccurate, and unconfident responses. The computational formula for PF1 was

$$PF1 = .33\ \text{Accuracy} + .53\ \text{Confidence} - .51\ \text{Reaction Time}$$

using standard scores for accuracy, confidence, and reaction timebased on the mean and variance of their distributions across all subjects.

ERPs were recorded from midline frontal, central, and parietal electrodes (Fz, Cz, and Pz; Jasper, 1958), referred to average mastoids, filtered digitally to a bandpass of .1 to 25 Hz, and decimated to a final sampling rate of 50 Hz. The prestimulus baseline (200 ms) was adjusted to zero to remove any DC offset. Vertical and horizontal electrooculograms (EOG) were also recorded. Across subjects, a total of 8184 ERPs were recorded. Epochs containing artifacts were rejected and EOG-contaminated epochs were corrected (Gratton, Coles, & Donchin, 1983). Furthermore, any trial in which no detection response or confidence rating was made by a subject was excluded along with the corresponding ERP.

RESULTS

Data Sample Construction

Within each block of trials, a running-mean ERP was computed for each trial. Each running-mean ERP was the average of the ERPs over a window that included the current trial plus the 9 preceding trials for a maximum of 10 trials per average. Within this 10-trial window, a minimum of 7 artifact-free ERPs were required to compute the running-mean ERP. If fewer than 7 were available, the running mean for that trial was excluded. Thus each running mean was based on at least 7 but no more than 10 artifact-free ERPs. This 10-trial window corresponds to about 30 s of task time. The PF1 scores for each trial were also averaged using the same running-mean window applied to the ERPs, excluding PF1 scores for trials in which ERPs were rejected.

Prior to analysis, the running-mean ERPs were clipped to extend from time zero (stimulus onset time) to 1500 ms post-stimulus, for a total of 75 time points. Sample running-mean ERPs (prior to application of rejection criteria) for one subject from one block of 50 trials are shown in Figure 1. Over the course of the block, complex changes in the shape of the ERP are evident.

The set of running-mean ERPs was split into a screening sample for building models and a calibration sample for cross-validation of the models. For each subject, odd-numbered blocks of trials were assigned to the screening sample, and even blocks were assigned to the calibration sample. After all trial-rejection criteria were satisfied, 2765 running-mean ERPs remained in the screening sample and 2829 remained in the calibration sample.

Linear Regression Models

A multiple-electrode (Fz, Cz, Pz) covariance-based PCA was performed on the running-mean ERPs. Each observation consisted of the 75 time points for each electrode for a total of 225 variables per observation. The BMDP program 4M (Dixon, 1988) was used for the calculations, using no rotation and extracting all factors with an eigenvalue greater than 1. One hundred and thirty-six factors were extracted, accounting for 99.45% of the variance in the data. The decay of the eigenvalues was roughly exponential, with the first 10 factors accounting for 70.96% of the variance in the data. Factor scores were computed for each running-mean ERP and stored for model development.

The DWT was computed using the same ERPs as in the PCA. A Daubechies analyzing wavelet (Daubechies, 1990) was used to compute the DWT of the EEG data over four scales. The length of the filters used for this wavelet was 20 points. This results in very smooth signal expansions in the wavelet transform. The scale boundaries and center frequencies of the scales are listed in Table 1.

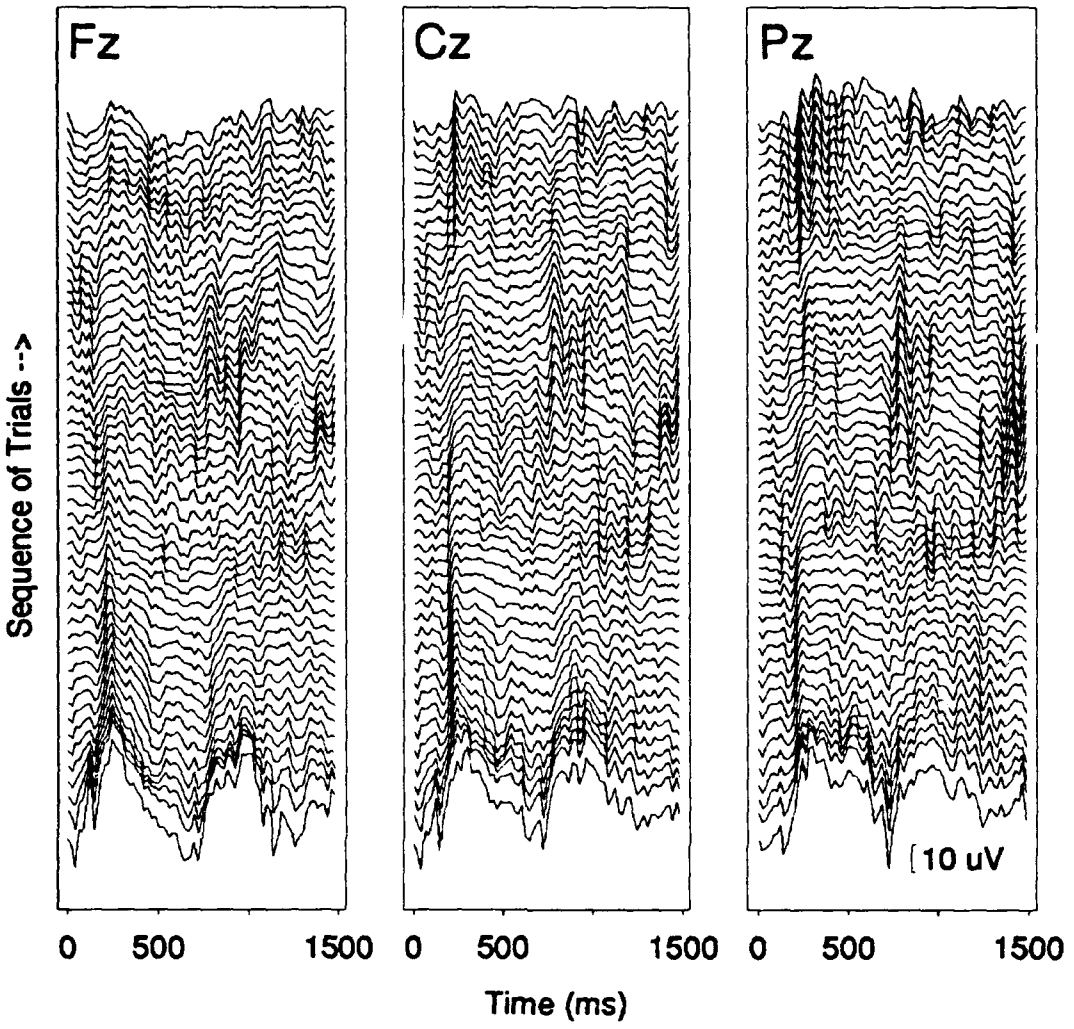


Figure 1. Running-mean ERPs at sites Fz, Cz, and Pz for subject 2 in the first block of 50 trials. Zero on the abscissa represents the stimulus onset (appearance of the display symbol used for the signal detection task). The ordinate represents scalp voltage at each electrode site; positive is up. The running-mean ERPs for successive trials of the block are stacked vertically from bottom to top (lowest is first).

The deciminated transform was centered within the ERP epoch (a factor of 2 at successive scales) yielding a total of 70 coefficients per transform (very low frequency scales and the DC term were excluded). The number of coefficients used was approximately halved with each increasing scale after decimation. For scales 0-3, the respective numbers of coefficients were 37, 19, 9, and 5. The real values of the DWT were stored for model development. No further transformations were performed.

Linear regression models for predicting performance (PF1), from either the PCA factor scores or from the DWT coefficients of the running-mean ERPs, were developed using a stepwise approach (BMDP program 2R). A criterion F-ratio of 4.00 was used to control the entry of predictor variables into a model. The F-ratio to remove a variable from a model was 3.99, resulting in a forward-stepping algorithm. The performance of each model was assessed by examining the coefficient of determination, r^2, as a function of the number of predictors entered (r^2 is the square of the multiple correlation coefficient between the data and the model predictions and also measures the proportion of variance accounted for by the model when the sample size is adequate and distributional assumptions are met).

Table 1. Scales of the 20-point Daubechies Discrete Wavelet Transform

Scale	Bandwidth (Hz)		Center Frequency (Hz)
0	10.50	25.00	16.20
1	4.42	10.50	6.82
2	1.86	4.42	2.87
3	0.78	1.86	1.20

Using the criteria described above, 90 factors of the PCA entered into models predicting PF1, and 92 coefficients of the DWT entered into models predicting PF1 (Figure 2). The r^2 increased for the PCA models in a fairly smooth, negatively accelerated fashion from a minimum of .07 for a single factor model to a maximum of .58 using 90 factors as predictors. The r^2 for the DWT model based on a single coefficient was .12, nearly double that of the PCA model based on a single factor. The increase in r^2 for the DWT models was almost linear for models using up to four coefficients as predictors; beyond that, further increases occurred in a piece-wise linear fashion reaching a maximum of .62 using 92 predictors. The greatest difference in r^2 between the DWT and PCA models (.11) also occurred with four predictors.

Prior experience has shown that models using more than 10 predictors have limited generality and are difficult to interpret. For this reason, cross-validation of the PCA and DWT models was performed with no more than 20 predictors. The models developed using the screening sample were applied in turn to the PCA scores and DWT coefficients of the calibration sample. As for the screening sample, performance of the models for the calibration sample was assessed using r^2 (Figure 3). In addition, the significance of r^2 was assessed using a F-ratio test (Edwards,

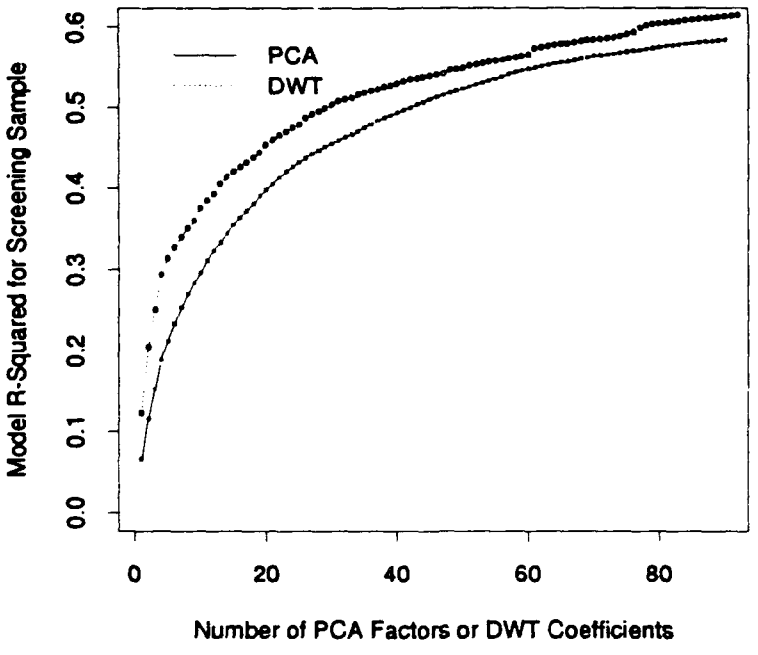


Figure 2. Coefficients of determination (r^2 or variance accounted for) for PCA and DWT models developed to predict task performance (PF1) for eight subjects in a signal detection task. Models were based on a screening sample of running-mean ERP and PF1 data, drawn from odd-numbered blocks of trials. Models are assessed by the r^2 as a function of the number of predictors entering into the model. Only models in which predictors met a criterion F-ratio of 4.0 to enter (3.99 to remove) are shown.

1976). This test used an adjusted number of degrees of freedom for the denominator, to allow for the serial correlation in the data introduced by computing the running means of the ERPs. In effect, the number of degrees of freedom was divided by 10, to allow for the 10-trial cycle length of the running-mean window. A conservative significance level of .001 was chosen, given the large number of models computed. The contour of r^2 values at this significance level appears as a dot-dashed line in Figure 3.

All of the PCA and DWT models tested explained significant proportions of variance in the calibration data set. For the PCA models, calibration r^2 rose gradually from a nearly insignificant level to a maximum of .22 using 10 predictors. The equation for the 10-predictor PCA model was

$$\begin{aligned} PF1 = {} & .11\,Factor2 - .10\,Factor4 + .13\,Factor5 - .05\,Factor8 \\ & - .09\,Factor9 + .08\,Factor11 - .06\,Factor15 - .08\,Factor43 \\ & + .07\,Factor47 - .07\,Factor68 + .02 \end{aligned}$$

where the factors are indexed according to the proportion of variance accounted for in the running-mean ERPs. The factor accounting for the greatest variance in the ERPs (Factor 1) did not enter the model. Five of the first 10 factors (Factors 2, 4, 5, 8, and 9) entered the model. Respectively, these factors accounted for proportions of variance in the ERPs of .12, .031, .0283, .0184, and .0169, or a total of .21 (21%). The entry of some of the higher factors in the 10-predictor model is surprising, given the small amount of variance in the ERPs that they account for. For example, Factors 11, 15, 43, 47, and 68 accounted for proportions of variance equal to .014, .01, .0022, .0019, and .0011, respectively, or a total of .0292 (under 3%).

Among the DWT models, the calibration r^2 for a single predictor (.11) was well above that of the corresponding single-factor PCA model (.04) and rose to a maximum of .22 using five DWT coefficients as predictors. The DWT coefficients are coded by electrode (Fz, Cz, Pz), scale (S0, S1, S2, S3) and time index (T0, T1, ..., TN). Actual latencies of the time points are obtained by multiplying the time index by 20 ms, the sampling period. The best single-predictor model was based on coefficient CzS3T22, with a regression coefficient of -.03 and an intercept of .02. Beyond five predictors, the r^2 for the DWT models declined slightly, and leveled off after about 10 predictors, showing no further

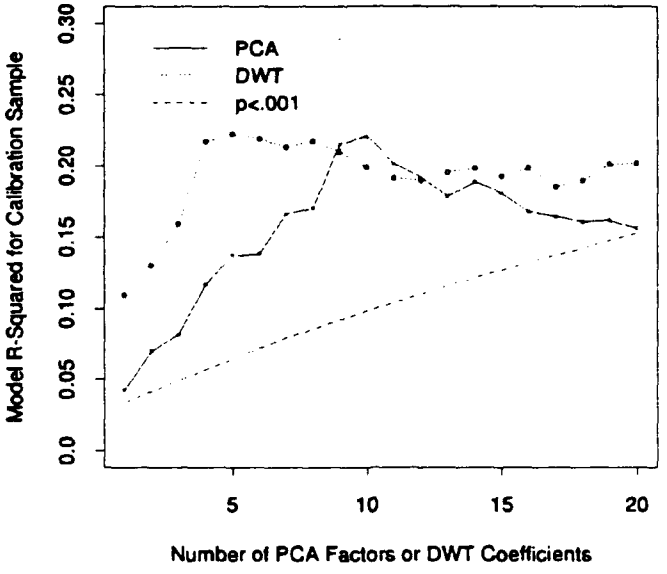


Figure 3. Coefficients of determination (r^2 or variance accounted for) for the first 20 PCA and DWT models of Figure 2, cross-validated using running-mean ERP and PF1 data from a calibration set of data drawn from even-numbered blocks of trials. The dot-dashed line indicates the contour of r^2 values significant using an F-ratio test at the $p < .001$ level where the numerator degrees of freedom depends on the number of predictors and the denominator degrees of freedom is one-tenth of the sample size. Values above this contour are significant.

improvement. As for the screening sample data, the greatest difference in r^2 between the DWT and PCA models for the calibration sample (.10) occurred with four predictors.

The equation of the best five-predictor DWT model selected by the stepwise regression algorithm was

$$PF1 = -0.03 * FzS2T6 + 0.04 * FzS2T22 + 0.06 * CzS2T6 - 0.05 * CzS2T22 - 0.05 * PzS2T6 - 0.17.$$

It is clear that a single scale, number 2, is most important for predicting task performance. This scale mainly reflects the time course of energy within the bandwidth of .078 to 1.86 Hz, which overlaps the range of the delta band of the EEG (1- 3.5 Hz) and will include some influence from low-frequency ERP components such as the P300 and slow waves. Two time intervals are indicated across electrodes: point 6 at Fz, Cz, and Pz (120 ms), and point 22 and Fz and Pz (440 ms). Frontal and parietal energy (Fz, Pz) in scale 2 at 120 ms is inversely related to PF1 as shown by the negative regression coefficients, whereas central activity (Cz) is positively related to PF1. Central and parietal energy (Cz, Pz) in scale 2 is inversely related to PF1 at 440 ms.

One potential problem with the wavelet analysis performed here stems from the length of Daubechies filters used (20 points). These filters had lengths over one fourth the length of the signals (75 points). While these filters produce smooth wavelets, they also increase the "support" of the transforms in the time domain. This means that the transforms are extrapolated in time before and after the interval of the signal. It is possible to decrease the support of the wavelet transform at the expense of smoothness by using shorter filters. To test the effects of shorter filters, the current data were partially re-analyzed using Daubechies filters of 4 points in length. With these filters, the support of the transform is reasonable at all four of the scales analyzed and time resolution of signal features at the larger scales is more precise than with the 20-point filters.

The most important single predictor for the 4-point filter DWT was located at electrode Cz and scale 2, as for the best single-predictor model based on 20-point filters. However, the wavelet coefficient in the 4-point filter model, CzS2T15, was at the 15th time point or a latency of 300 ms. This lies 120 ms earlier than the scale 2 coefficient in the best single-predictor model based on the 20-point filters (CzS2T22). The regression coefficient for CzS2T15 in the 4-point filter model was .03, with an intercept of -.16. This regression coefficient is negative, whereas the regression coefficient for CzS2T22 in the 20-point filter model was positive. The difference in sign suggests that CzS2T15 in the 4-point filter model is a different feature of the ERP than CzS2T22 in the 20-point filter model, even though it is in the same scale and at the same electrode. The cross-validation r^2 for the 4-point filter based on CzS2T15 was .15, which is higher than the r^2 for CzS2T22 in 20-point filter model (.11).

Neural Network Analyses

In addition to the linear regression models, feed-forward artificial neural networks were trained using the backpropagation method (Rumelhart & McClelland, 1986) to predict PF1 from ERP patterns. Three networks were trained: 1) raw ERPs; 2) PCA scores; and 3) DWT coefficients. For the ERP network, the inputs were the voltages in the ERP time series for electrodes Fz, Cz, and Pz. These were the same data used to derive the PCA scores and DWT coefficients used in the linear regression models described earlier. There were 75 points per electrode spanning a latency range of 0-1500 ms, for a total of 225 network inputs. For the PCA network, the PCA scores used in the linear regression models described above served as inputs. As for the linear regression models, only the first 136 factors were retained.

Table 2. Scales of the 4-point Daubechies Discrete Wavelet Transform

Scale	Bandwidth (Hz)		Center Frequency (Hz)
0	10.88	25.00	16.49
1	4.74	10.88	7.18
2	2.06	4.74	3.12
3	0.90	2.06	1.36
4	0.39	0.90	0.59

For the DWT network, three changes were made in the generation and selection of DWT coefficients. First, the wavelet transform was based on the 4-point Daubechies filters which appeared to be superior to the 20-point filters used in the initial linear regression models. Second, since low frequency information seemed valuable in the linear regression models, the range of the transform was extended, adding a fifth scale (Table 2). Third, selection of the coefficients was not performed by the decimation approach taken for the linear regression models. Instead, the undecimated transforms were computed (Shensa, 1991), yielding 75 points for each scale. Then the mean power of each coefficient was computed and the top 20% of the coefficients at each scale were selected as inputs to the network (Figure 4). This resulted in a set of 225 coefficients, or about the same number that would be obtained by decimation. However, this scheme selects coefficients that are high in power, and so account for large proportions of the ERP signal variance at each scale.

Networks were trained and tested with a commercial software package (Brainmaker, California Scientific Software, Inc.). All three networks consisted of two layers. A single "hidden" layer consisting of three neurons received connections from all the inputs. These three neurons were fully connected to the output layer, which consisted of a single neuron. The teaching signal for this neuron was PF1. In addition to inputs from other neurons, each neuron received a constant "bias" input, which was fixed at a value of 1.0.

The output transfer function for all neurons was the logistic function with a gain of 1.0 and a normalized output range of 0.0 to 1.0. The learning rate was 1.0 and the momentum was 0.9. All inputs and the desired output (PF1) were independently and linearly normalized to have a range of 0.0 to 1.0. As for the linear regression models, the screening sample (half the runs) was used for training the networks and the calibration sample (the remaining runs) was used for testing. Training proceeded by adjusting the connection weights of the neurons for every input vector. The training tolerance was 0.1, i.e, if the absolute error between the network output (predicted PF1) and the actual PF1 value for a trial exceeded 10%, then the connection weights were adjusted using the backpropagation algorithm.

Prior to training, the sequence of input vectors was randomized. Training involved repeated passes (training epochs) through the screening sample and was stopped after a maximum of 1000 training epochs. Testing was performed on the calibration sample at intervals of 10 training epochs. The validity of a trained network was measured in terms of the proportion of calibration sample trials for which PF1 was correctly predicted to within the criterion 10% margin of error. The curve relating the proportion of correctly predicted calibration sample trials to the number of training epochs will be referred to as the *generalization learning curve* (Figure 5).

The probability of correctly guessing a uniform random variable with a range of 0.0 to 1.0 with a 10% margin of error is 0.2. As shown in Figure 5, two of the three networks trained to predict PF1 in the calibration sample better than 0.2 with as few as 10 training epochs. Beyond 50 training epochs, the generalization learning curves of the three networks begin to diverge.

The DWT network appears to "learn" to generalize about as well as it can by about 290 training epochs. For this network, the proportion correct jumps from about 0.25 to over 0.3 near 200 epochs. From that point on, a rough plateau in the curve is held, with a few dips between 800 and 1000 epochs. The maximum proportion correct of 0.348 occurs at epoch 930, but this is not substantially (or significantly) greater than an earlier maximum of .346 at epoch 290.

For the ERP network, a gradual rise in the proportion correct occurs between 10 and 400 epochs, reaching a maximum of 0.331 at training epoch 350. Beyond 400 epochs, the proportion correct for the ERP network declines gradually to near chance levels of performance.

The generalization learning curve of the PCA network exhibits the most complex shape, rising and falling repeatedly over the 1000-epoch range. Interestingly, it also shows a large step near 200 epochs, as did the DWT network, and an early maximum of 0.279 at 250 epochs, after which the curve declines and oscillates up to about 850 epochs. At that point the curve rises again, reaching a new, higher maximum of 0.288 at 940 epochs.

Although the curves in Figure 5 are complex, two generalizations seem possible. First, within the 1000-epoch scope of the training, all three networks appear to achieve near-maximal levels of generalization performance within the first 400 training epochs. Beyond 400 training epochs, further training appears to produce either declines or oscillations in generalization performance, and only small increases above the earlier maximum proportions of correctly predicted trials occur. Second, the DWT network trained most rapidly and achieved the highest and most stable level of generalization performance. The DWT network "learned" to generalize to new data faster than the ERP network by about 60 training epochs.

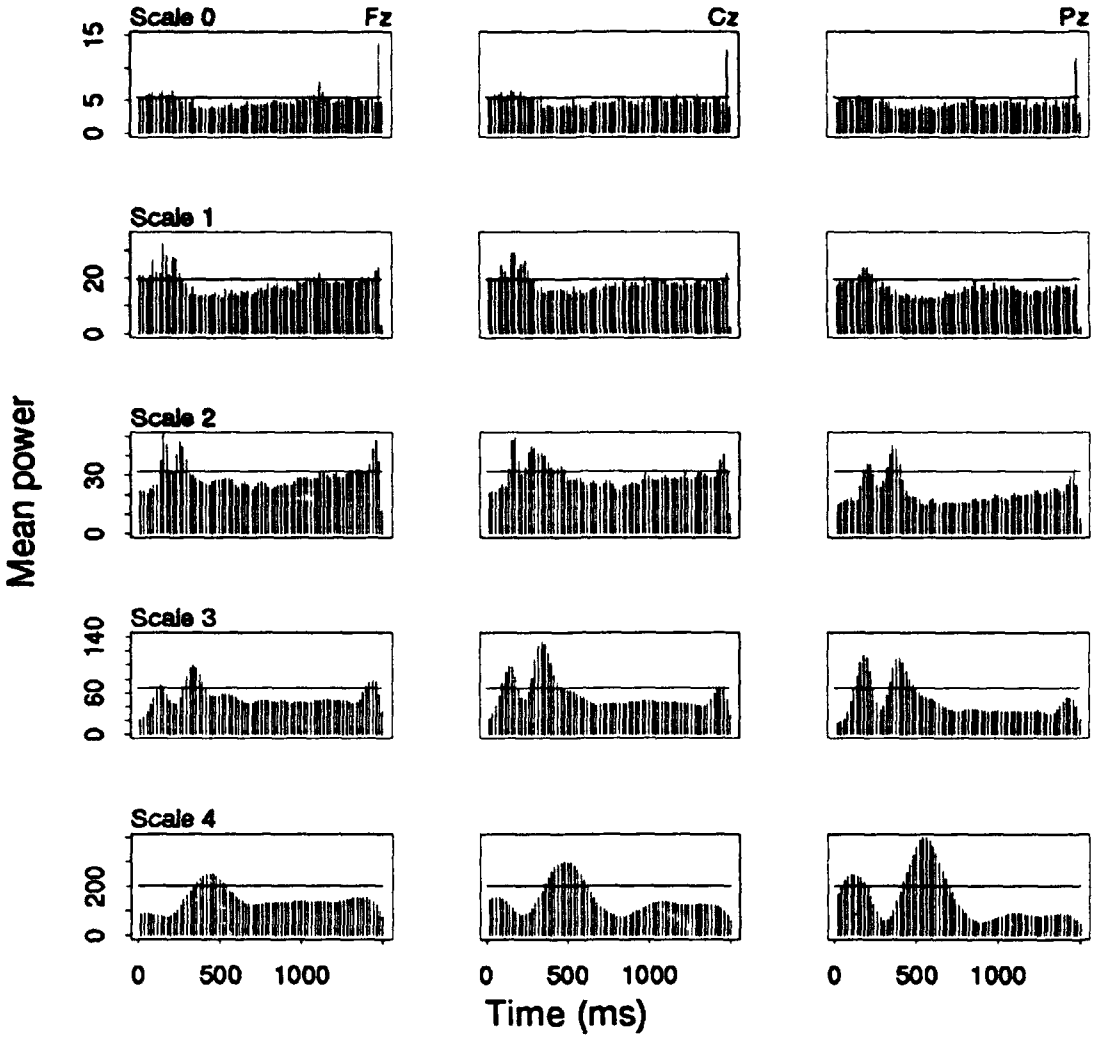


Figure 4. Mean power of the undecimated 5-scale DWT coefficients at electrodes Fz, Cz, and Pz, used for the neural network trained to predict PF1. The DWT coefficients for each running-mean ERP were squared, summed, averaged and plotted as a function of time relative to the stimulus. Each row of graphs represents one scale of the transform beginning with the smallest scales at the top (see Table 2) and proceeding to the largest scale at the bottom. Each column of graphs corresponds to one electrode site in the order Fz, Cz, Pz, from left to right. The 80% quantile was computed across electrodes within each scale and is shown by the horizontal line in each graph. Coefficients with mean power values greater than the 80% quantile, i.e., the top 20%, were used as inputs to the neural network.

The raw ERP network achieved a proportion correct approaching that of the DWT network (.331 versus .348) but was not as stable. A z test of the significance of the difference between these proportions based on the standard normal distribution (Fleiss, 1981, p. 23) yielded a p-value of 0.21. However, an F-test of the ratio of variances of proportions correct for the ERP and DWT networks between epochs 200 and 1000 rejected the hypothesis that the variances were equal (the alternative hypothesis was that the true ratio of variances was greater than 1.0), $F(79,79) = 3.12$, $p < 0.000$.

Generalization performance of the PCA network was lower than both the ERP and DWT networks. The z tests of the differences between the proportions correct of DWT and PCA networks and of ERP and PCA networks yielded p-values of 0.0015 and 0.0162, respectively.

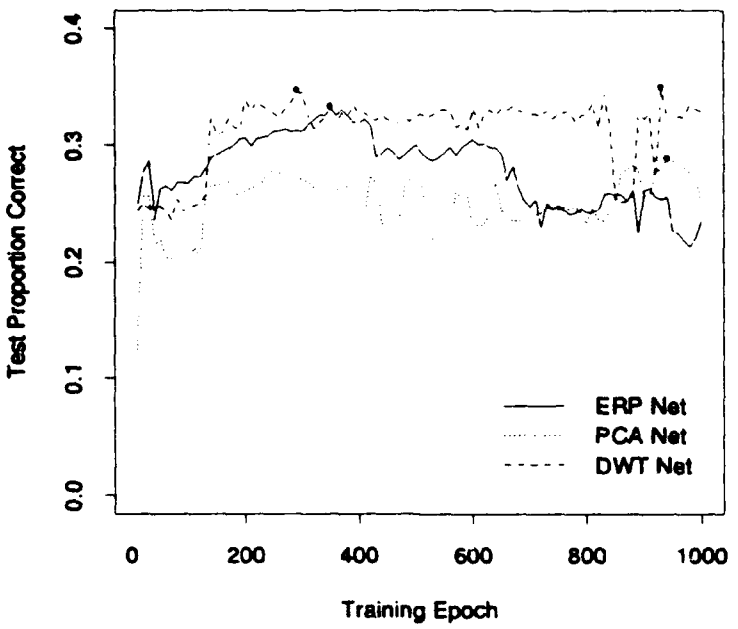


Figure 5. Generalization learning curves of the three neural networks trained to predict PF1 from raw ERPs (solid line), PCA scores (dotted line), or high-power DWT coefficients (dashed line). The abscissa marks the number of training epochs (complete passes through the screening sample) and the ordinate marks the proportion of trials in the calibration sample for which PF1 was correctly predicted with a 10% margin of error. The solid circles mark the highest proportion correct for each network.

Decorrelation and Energy Compaction

Statistical independence of the predictor variables could be one reason why the linear regression models based on PCA scores and the DWT were more successful than the peak and latency measures used in earlier analyses. In the signal processing literature, the tendency of a transform to render independent measures from multivariate data is called *decorrelation*. Decorrelation efficiency compares the sum of the off-diagonal terms in the covariance matrices of the original (raw ERPs) and the transformed data (Akansu & Haddad, 1992, p. 28). A transform that perfectly decorrelates the data has a decorrelation efficiency of 1.0.

The decorrelation efficiency of the 4-scale DWT used here was 0.13. Although the factors obtained with PCA are decorrelated, the factor scores which represent the data may be correlated. For this reason, the decorrelation efficiency of the PCA, measured from the covariance matrix of the factor scores was not 1.0, but .64, which is still several times higher than the decorrelation efficiency of the DWT. However, the DWT regression models explained the same amount or more variance in the data using fewer variables than the PCA models. Thus it appears that the degree of decorrelation of a transform alone does not determine how well it will extract important ERP features for modeling task performance.

The relatively small number of DWT coefficients needed to generalize to new data using linear regression models suggests that the DWT efficiently extracts a small but behaviorally important set of features from the ERP. The relative speed of generalization learning by the DWT neural network may also be consistent with this idea. If only a small proportion of the inputs contain information related to the output then only the weights corresponding to those inputs would require adjustment, leading to faster generalization learning.

In signal processing, the property of a transform that describes its tendency to concentrate information in a small proportion of the variables is called *energy compaction* (Akansu & Haddad, 1992, p. 28). Good energy compaction means having a small number of large values on the diagonal of the covariance matrix of the transform variables. It is measured as a function of the number of variables retained to fit the data, sorted in order of decreasing covariance.

Energy compaction could also result in more robust models, showing less over-fitting. This could occur when the variables that explain most of the variance enter first, leaving only variables of low influence to adversely affect the fits when added later.

For the data used in the linear regression models, energy compaction measures of the raw ERPs, PCA scores, and DWT coefficients for 5 variables was .06, .08, and .09. For 10 variables, energy compactions for ERP, PCA, and DWT were .11, .15, and .16, and for 20 variables, energy compactions were .20, .25, and .26, respectively. Thus over the range of models tested, the DWT was only slightly more efficient in compacting the energy (or variance) in the data than the PCA. It seems unlikely that such small differences in energy compaction (about 1%) could account for the higher efficiency of the DWT models than the PCA models.

DISCUSSION

Linear Regression Models

Both PCA and DWT methods yielded linear regression models that significantly explained signal detection performance in a 30 s running window and generalized to novel data. Both methods also performed better than a traditional peak amplitude and latency analysis of the running-mean ERPs. For comparison, the best stepwise linear regression model developed using predictors drawn from a set of 96 multi-electrode amplitude and latency measures of the ERP on the same data set yielded an r^2 of .28 for the screening sample and failed to significantly cross-validate on the calibration sample (Trejo & Kramer, 1992; peak amplitude- and latency-based models did cross-validate when adapted to the ERP waveforms of individual subjects).

The DWT models were clearly superior to the PCA models when based on a small number of predictors. Twice as many PCA factors were required to explain the same amount of variance in the data as DWT models based on 5 coefficients. In cross-validation, no advantage of the PCA models over DWT models was evident with any number up to 20 predictors. The PCA models showed evidence of over-fitting the data when more than 10 predictors were used, as shown by the decline in r^2 for the calibration sample for models using 10 to 20 predictors. In contrast, the DWT models suffered relatively small decreases in r^2 when using more than 5 coefficients.

Single-predictor models for the DWT based on 4-point filters were compared to the 20-point filters used initially to determine the sensitivity of the location estimates to filter length. The net effects of using shorter filters to compute the wavelet transform were to change the location estimate, but not the electrode or scale estimates of the best single predictor model, and increased cross-validation r^2. The higher cross-validation r^2 for the 4-point filter model than the 20-point filter model was unexpected. However, this result suggests that more precise temporal localization of features in the wavelet transform may provide more robust representation of the ERP or EEG features associated with task performance.

PCA is known to produce factors that resemble the shape and time course of ERP components. The information provided by the DWT is somewhat different. For example, the 5-predictor DWT model indicated that a pattern of energy at specific latencies in the ERP confined to the bandwidth associated with P300, slow waves, and EEG delta band activity, was correlated with signal detection performance across a sample of eight subjects. It is well known that P300 and slow waves co-vary with the allocation of cognitive resources during task performance. However, it is not clear whether the wavelet coefficients included in the regression models are simply better measures of P300 and slow wave or if they represent new aspects of the ERP. Comparisons of ERPs reconstructed from the DWT coefficients and the average ERP waveforms will be required to express the coefficients in terms of familiar ERP peaks.

Neural Networks

As for the linear regression models, the best generalization performance of neural networks — measured in terms of predicting PF1 in the calibration sample — was achieved with the DWT representation of the ERPs. Somewhat surprisingly, neural networks trained to predict PF1 from raw ERPs generalized almost as well as the DWT. Both ERP and DWT-based networks generalized to new data significantly better than networks based on PCA scores.

The neural network based on the DWT required fewer training epochs than the raw-ERP network to reach a maximal level of generalization to new data. In addition, beyond the initial training period of 200 epochs, generalization performance of the DWT network was more stable than the ERP network. After about 400 training epochs, the generalization learning curve declined for the ERP network, indicating over-fitting of the data in the

screening sample. In contrast, the generalization learning curve for the DWT exhibited a few dips, but remained surprisingly stable over most of the training range, indicating a resistance to over-fitting. This result agrees with the resistance to over-fitting observed with more than the optimum number coefficients in the linear regression models based on the DWT.

General Conclusions

The results described here show that the DWT can provide an efficient representation of ERPs suitable for performance-prediction models using either linear regression or neural network methods. Furthermore, the DWT models tested here needed the fewest parameters, exhibited highest generalization and were relatively insensitive to the detrimental effects of over-fitting as compared to models based on PCA scores or raw ERPs. This result, together with the initial rise in r^2 for the linear regression DWT models (Figure 3) suggest that the DWT coefficients measure unique and important sources of performance-related variance in the ERP.

The superiority of the DWT over PCA seen in the models tested here cannot be explained in terms of decorrelation and energy compaction properties of these transforms. Decorrelation was actually higher for PCA than for the DWT, and energy compactions over the range of variables included in the models were about equal for the two transforms. Instead, it appears that the DWT simply provides more useful features than PCA, when utility is measured by how efficiently task performance can be predicted using ERPs.

For practical ERP-based models of human performance, ease of model development and speed of computation are also important factors. The cost of computing the DWT is trivial when compared to deriving a PCA solution, which involves inverting and diagonalizing a large covariance matrix. Even more time is required for peak and latency analyses, which depend on expert human interpretation of the waveforms.

The nature of the features extracted using the DWT merits further study. By identifying the time and scale of energy in the ERP related to task performance, specific ERP or EEG generators may be indicated, as shown by the dominant presence of slow waves and delta-band activity in the 5-predictor linear regression DWT model of signal detection performance. In this way, the DWT may provide new insight into the physiological bases of cognitive states associated with different performance levels in display monitoring tasks.

Future work should examine the reconstructed time course and scalp distribution of the patterns indicated by DWT or other wavelet models and relate these to known physiological generators. Through inversion of the DWT, it is possible to reconstruct the time course of the energy indicated by a specific model. In addition, other wavelet transforms may provide a finer analysis of the time-frequency distribution of the ERP. For example, wavelet transforms using multiple "voices" per scale, such as the Morlet wavelets or wavelet packets, provide much finer resolution than that afforded by orthonormal wavelets used in this study. In addition, data from other kinds of tasks should be analyzed and the development of models for individual subjects should be also explored.

REFERENCES

Akansu, A. N., & Haddad, R. A. (1992). *Multiresolution Signal Decomposition. Transforms, Subbands, and Wavelets.* San Diego: Academic Press.

DasGupta, S., Hohenberger, M., Trejo, L. J., & Mazzara, M. (1990). Effect of using peak amplitudes of ERP signals for a class of neural network classification. *Proceedings of the First Workshop on Neural Networks: Academic / Industrial / NASA / Defense*, pp. 101-114, Auburn, AL: Space Power Institute.

DasGupta, S., Hohenberger, M., Trejo, L., & Kaylani, T. (1990, April). Effect of data compression on ERP signals preprocessed by FWT algorithm upon a neural network classifier. *The 23rd Annual Simulation Symposium, Nashville, TN.*

Daubechies, I. (1990). The Wavelet Transform, Time-Frequency Localization and Signal Analysis. *IEEE Transactions on Information Theory, 36 (5),* 961-1005.

Daubechies, I. (1992). *Ten Lectures on Wavelets.* Philadelphia: Society for Industrial and Applied Mathematics.

Dixon, W. J., (1988). *BMDP Statistical Software Manual.* Berkeley: University of California Press.

Edwards, A. L. (1976). *An Introduction to Linear Regression and Correlation.* San Francisco: W. H. Freeman and Co.

Fleiss, J. L. (1981). *Statistical methods for rates and proportions.* Second edition. New York: John Wiley and Sons.

Gratton, G., Coles, M. G., & Donchin, E. (1983). A new method for off-line removal of ocular artifact. *Electroencephalography and Clinical Neurophysiology, 55,* 468-484.

Humphrey, D., Sirevaag, E., Kramer, A. F., & Mecklinger, A. (1990). *Real-time measurement of mental workload using psychophysiological measures.* (NPRDC Technical Note TN 90-18). San Diego: Navy Personnel Research and Development Center.

Jasper, H. (1958). The ten-twenty electrode system of the international federation. *Electroencephalography and Clinical Neurophysiology, 43,* 397-403.

Kaylani, T., Mazzara, M., DasGupta, S., Hohenberger, M., & Trejo, L. (1991, February). Classification of ERP signals using neural networks. *Proceedings of the Second Workshop on Neural Networks: Academic / Industrial / NASA / Defense,* pp 737-742. Madison, WI: Omnipress.

Rumelhart, D. E., & McClelland, J. L. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition. Vol. 1. Foundations.* Cambridge, MA: MIT Press/Bradford Books.

Ryan-Jones, D. L. & Lewis, G. W. (1991, February). Application of neural network methods to prediction of job performance. *Proceedings of the Second Workshop on Neural Networks: Academic / Industrial / NASA / Defense,* pp. 731-735. Madison, WI: Omnipress.

Shensa, M. J. (1992). The discrete wavelet transform: Wedding the á trous and Mallat algorithms. *IEE Transactions on Signal Processing, 40,* 2464-2482.

Trejo, L. J., Lewis, G. W., & Kramer, A. F. (1991). ERP indices of human performance: Effects of stimulus relevance and type of information processing. Society of Psychophysiology, 31st Annual Meeting, Chicago, October 9-13, 1991.

Trejo, L. J., & Kramer, A. F. (1992). ERP indices of performance quality in signal detection, running memory, and computation. American Psychological Society, Annual Convention, June 20-22, San Diego.

Tuteur, F. B. (1989). Wavelet transformations in signal detection. In J. M. Combes, A. Grossman, & P. Tchamitchian, (Eds.), *Wavelets: Time-frequency methods and phase space.* New York: Springer-Verlag, pp. 132-138.

Venturini, R., Lytton, W. W., & Sejnowski, T. J. (1992). Neural network analysis of event related potentials and electroencephalogram predicts vigilance. In J. E. Moody, S. J. Hanson, and R. P. Lippmann (Eds.), *Advances in Neural Information Processing Systems 4,* pp. 651-658. San Mateo, CA: Morgan Kaufmann Publishers.

Neural Network Discrimination of Brain Activity

David L. Ryan-Jones & Gregory W. Lewis
Navy Personnel Research and Development Center
San Diego, CA 91252-7250

Electroencephalogram (EEG) and event-related brain potential (ERP) data are often recorded in studies of brain processing (Gevins, 1986; Regan, 1988). EEG and ERP records are samples over time of the electrical activity resulting from changes in polarization of the neurons in the brain. ERP data differ from EEG data in that ERPs are sampled relative to the time of onset of a specific stimulus. An sample of electrical activity over time is known as an epoch. Single-epoch ERP samples contain brain responses associated with the sensation and perception of a stimulus, cognitive activity, and the behavioral preparations of the subject. Although single-epoch ERP data may be desirable in the analysis of the brain processing related to the stimulus event, single-epoch waveform features are small compared to the more random background EEG activity. Single-epoch ERP data for two subjects are shown in Figure 1. Note that single samples of activity within subjects and between subjects do not look exactly alike. As a result, several epochs are usually averaged to reduce the background noise level. The number of epochs required for the average waveform is related to the average amplitude of the specific feature of interest. Average ERPs for the data in Figure 1 are shown Figure 2. As with the single-epoch ERP data, the average ERPs look very different from each other, and from the single-epoch samples that make up the averages.

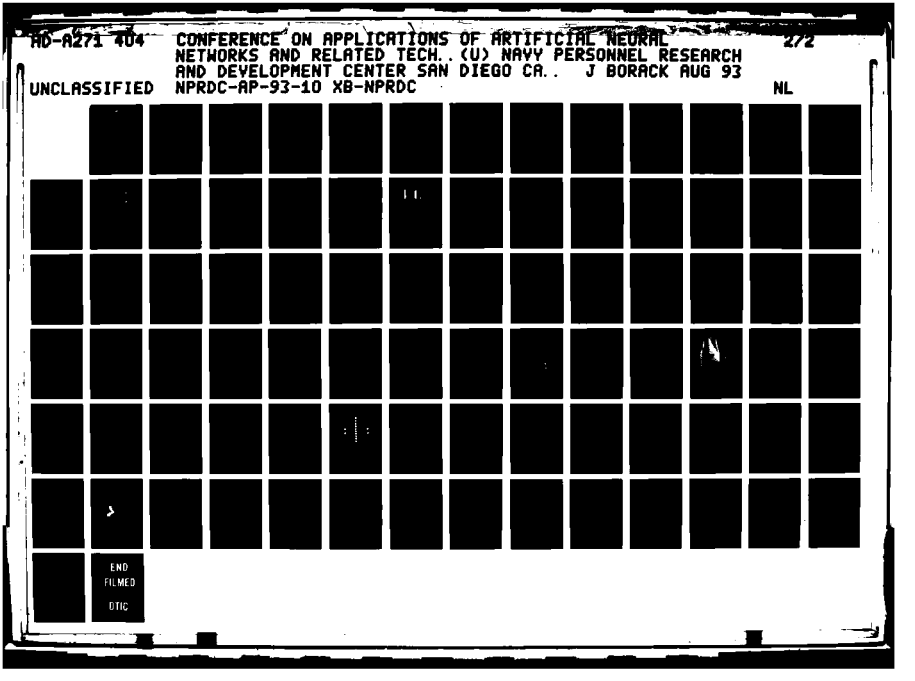
Another factor to be considered with single-epoch data is that ERP features are generated by complex nonlinear processes. As a result, single-epoch waveform features may have unknown distributions, and different features may have very complex interrelationships. Therefore, traditional statistical techniques may not be the best method way to analyze single-epoch data. In spite of these problems, single-epoch ERP analysis may be desirable, especially when few samples are available or when real-time processing of data is desired. Recently, neural network analysis techniques have been used extensively to extract weak signal features masked by noise, and to interpret data containing unknown nonlinear relationships (Anderson, S., 1990; Benediktsson, Swain, & Ersoy, 1990). The purpose of the current research was to determine whether neural network techniques could be used to improve the analysis and interpretation of single-epoch ERP data.

The data used in this study were derived from a visual discrimination experiment. Five male subjects were required to make discriminations between geometric designs and human faces. There were a total of 5 different designs, and each design was presented 8 times. There were 5 different faces, and each face was presented 8 times. Thus, there were a total of 40 geometric stimuli, and 40 human faces. The stimuli were displayed in random order on a monochrome CRT with an average interstimulus interval of 3 sec. During performance of the task, ERP data were sampled from sites F3, F4, C3, C4, P3, P4, O1, and O2 referenced to linked mastoids. Each epoch consisted of 125 ms prestimulus period, and an 825 ms poststimulus period. The data were sampled at 128 Hz, 20K gain, and filtered with a bandpass of 0.1-100 Hz bandpass. The single-epoch data for each type of stimulus (geometric design or face) for each subject were then averaged, and comparisons were made between the recognized waveform features generated by the two types of stimuli. No significant differences were found between the two stimuli in terms of the latency or amplitude any ERP waveform features. The single-epoch ERP data were not processed further, and EMG and eyeblink artifacts were not removed from the data. Artifacts were not removed from the data to simulate the conditions that would be encountered in the real-time application of the techniques. The 400 single-epoch ERPs (5 subjects x 80 ERPs/subject) were divided equally into two sets. The 200 odd-numbered epochs were used for the training set in the study, and the 200 even-numbered epochs were used for the test set. This selection rule was used instead of random selection to minimize changes in the ERP data due to adaptation over time on task.

A backpropagation learning algorithm was selected as the classification tool for the study. This algorithm was selected for several reasons. First, the mathematics and statistics of the backpropagation algorithm are well understood (Wan, 1990; White, 1989). Second, the backpropagation algorithm has

AD-A271 404 CONFERENCE ON APPLICATIONS OF ARTIFICIAL NEURAL NETWORKS AND RELATED TECH.. (U) NAVY PERSONNEL RESEARCH AND DEVELOPMENT CENTER SAN DIEGO CA.. J BORACK AUG 93 272

UNCLASSIFIED NPRDC-AP-93-10 XB-NPRDC NL



END
FILMED
DTIC

been shown to be very successful in classifying data with complex nonlinear relationships (Maren, Harston, & Pap, 1990). Third, many different backpropagation network packages are available in the commercial market, making it unnecessary to locally develop the software code to implement the algorithm. An IBM/PC compatible version of a commercial package was selected for this study. However, all of commercial software packages have some limitations in their implementation. In this case, the size of the data file was limited by the capacity of the file editor, and by the number of processing nodes allowed by the memory model. In order to meet these size limitations, the data were restricted to samples from two sites P4 and F3. Sites P4 and F3 were selected because previous studies of the ERPs generated by human faces have suggested that some of the waveform features at these sites exhibit an enhanced response to face stimuli compared to other kinds of stimuli. In particular, a positive wave at about 300 ms (P3), and a negative wave at about 500 ms (negative slow-wave or NSW) after stimulus onset have been associated with facial recognition. Thus, it was possible that differences in single-epoch data might be found where differences in averaged data had been previously noted. Keep in mind, however, that the amplitude and latency of these two waveform features were not significantly different for the two types of stimuli. In considering these data, one possible reason for this failure is the large trial-to-trial variability seen in the amplitude, latency, and shape of these waveform features.

One critical step in performing a network analysis is to define the parameters of the network. An unsuccessful network may only mean that the network parameters are not optimal for the problem at hand. Optimal network parameters can be determined, if one has the time, by comparing the performance of alternative forms. In this study, the optimal network consisted of 3 layers, including an input layer with 256 nodes, one hidden layer with 128 nodes, and an output layer with 1 node (face-not face). The input data consisted of 256 values (128 values from each sample recorded at P4 and F3). The data were normalized to values between 0 and 1, and the output value was allowed to vary between 0 and 1. The purpose of the data transform and output value restriction was to simplify the interpretation of the resulting network. The network was trained using odd-numbered epochs, and subsequently tested for generalization using the even-numbered epochs. Relationships between the variables at different layers in the network were modeled by the logistic transform function. The network learning rate was initially set to 0.9, and the rate was allowed to decrease to 0.7 as training progressed. Network weights were modified after each training example was processed by the network.

There are several different strategies that can be used to train a backpropagation network, including training to a fixed performance level, and training to the minimum least squared error. In this study, training continued until all of the epochs in the training set were correctly classified during a single pass through the data. During training, correct classification was considered to be an output of 0.9 and above for a face, or 0.1 and below for a geometric design. During testing, these values were lowered to 0.6 and above for a face, and 0.4 and below for a geometric design. Network training required about 72 hours of time on an IBM/PC 386-25 compatible computer. This translates into 2000 passes through the training set to realize the required 100% correct classification criterion. After training, the network was evaluated using the test set of ERP samples. The trained network was able to correctly classify 178 (89%) of the epochs in the test set (X^2=122.12, df=1, p<0.0001).

The results demonstrate that neural network techniques can be successfully applied to single-epoch ERP analysis. Of course, this was not entirely unexpected since these techniques have been successfully applied to data in other domains with similar characteristics. One problem with using these techniques is interpreting what the resulting network means within the context of traditional ERP analysis. The pattern of weights in the hidden layer of the network suggests that the neural network compared information about the amplitude and latency of the P3 and NSW components of the single-epoch ERPs. Remember, however, that there were no significant differences in the analysis of variance between the two stimuli in P3 or NSW amplitude or latency. The success of the neural network techniques in evaluating ERP data is not surprising since the backpropagation network can effectively utilize the nonlinear relationships in the data. One very important implication of this application of neural network techniques is that brain electrical activity can now be accurately interpreted in near real-

time. Thus, it may soon be practical to directly utilize brain activity to modify or control the characteristics of a man-machine interface.

REFERENCES

Anderson, S. (1990). Neural network applications of signal processing, pattern recognition, and real time intelligent processing. In *Proceedings of the First Workshop on Neural Networks: Academic/Industrial/NASA/Defense*. Madison, WI: Omnipress.

Benediktsson, J., Swain, P., & Ersoy, O. (1990). Neural network approaches versus statistical methods in classification of multisource remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, **28**, 540-552.

Gevins, A. (1986). Quantitative human neurophysiology. In H. Hannay (Ed.), *Experimental techniques in human neuropsychology*. New York: Oxford University Press.

Maren, A., Harston, C., & Pap, R. (1990). *Handbook of neural network computing applications*. San Diego: Academic Press.

Regan, D. (1988). *Human brain electrophysiology: Evoked potentials and evoked magnetic fields in science and medicine*. London: Chapman & Hall.

Wan, E. (1990). Neural network classification: A bayesian interpretation. *IEEE Transactions on Neural Networks*, **1**, 303-305.

White, H. (1989). Neural-network learning and statistics. *Neural Computing*, **1**, 425-464.

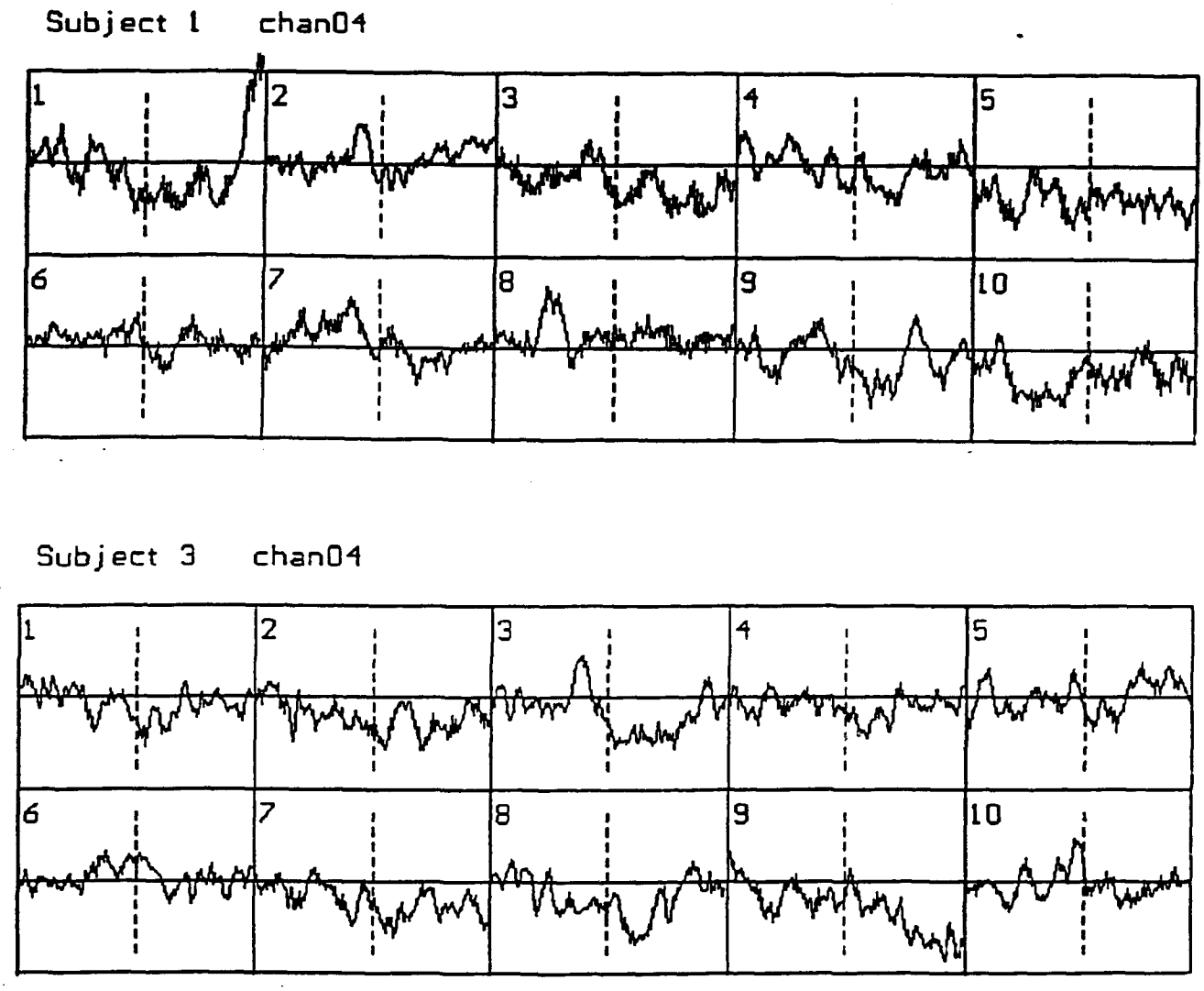


Figure 1. Single-Epoch ERP data for two subjects.

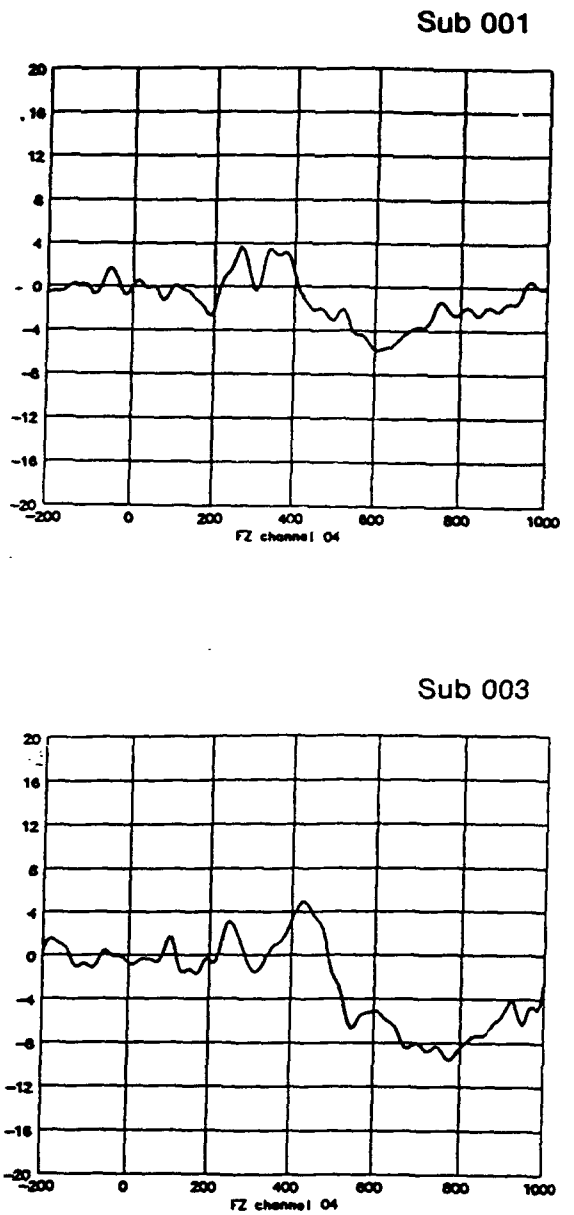


Figure 2. Averaged ERP data for two subjects.

Task Response Prediction Using Cortical Brain Potentials: A Neural Network Analysis

Gregory W. Lewis and David L. Ryan-Jones
Neurosciences Division
Training Research Department
Navy Personnel Research and Development Center
San Diego, California 92152-7250

BACKGROUND

Individual identification, and assessment are necessary for many purposes in our society. For example, it is important to control access to property, and equipment. Currently, access to secured areas, and computer systems has depended primarily upon security badges, and passwords. Both of these security methods can easily be subverted. New techniques are being investigated and developed for use as improved access control techniques. In addition to the traditional fingerprint, palm prints and photographs of the retina of the eye have also been used for identification purposes. Biochemical systems such as genetic testing are being increasingly used for forensic testing, but are not yet practical as an applied identification tool at this time. Systems which rely on anatomical features, as in the case of fingerprints, may be subverted. The approach described in this paper will improve upon the currently used techniques for individual identification.

Brain Recordings

For more than 50 years, the research literature has suggested that there are large individual differences in the electrical and magnetic activity in the brain. There is also evidence that some of the characteristics of brain activity may be stable when measured from day-to-day. Brain responses to sensory stimulation (e.g. visual, auditory, somatosensory, olfactory, gustatory) as well as higher-order cognitive processing (e.g., decision-making), now can be examined in great detail using a variety of recording procedures. An on-going record of brain electrical activity is called an electroencephalogram (EEG), and a comparable record of magnetic activity is called a magnetoencephalogram (MEG). EEG and MEG records usually have a great deal of uncontrolled variation, and special techniques are necessary to stabilize activity patterns. Brain activity can be stabilized by strict control of the conditions under which brain activity is generated. When human sensory systems are stimulated by an event such as a flash of light or a tone, there is a predictable sequence of processing that occurs in the brain. This processing generates an event-related potential (ERP) that can be recorded from the scalp beginning shortly after the onset of stimulation, and lasting for 1-3 seconds after the stimulation. These ERPs can be repeatedly generated from individuals who are given the same stimulus. Due to the low amplitude of the signal, it is often necessary to

repeatedly sample the response to the stimulus, and to average the response patterns. ERP measures are in the microvolt volt range (uV, millionths of a volt).

Comparable records of averaged magnetic activity are called evoked fields or event-related fields (ERFs). Neuromagnetic measures have only recently been possible. Due to the low amplitude of the signal, special low-temperature systems are required to measure the magnetic signals emitted by brain tissue. The unit of measurement is femtoTesla (10^{-15} Tesla). Neuroelectric and neuromagnetic recordings are subsets of more general measures, called bioelectric and biomagnetic measures. Bioelectric and biomagnetic measures refer to recordings from all types of tissue including neural, muscle, heart, and lungs.

ERP Stability

In the NPRDC laboratory, ERP recordings have been shown to be remarkably stable and unique to individuals (Naitoh & Lewis, 1981; Lewis, 1984; Lewis & Ryan-Jones, 1992). Although the actual shape of an ERP varies considerably from individual to individual, there is stability within individuals over time for individual waveforms. Sources of ERP variation include individual differences in brain anatomy, and differences in the way in which information is processed by the individual. Given these observations, it is now possible that ERP waveforms could be used as classifiers for several purposes. First, since ERP morphology is relatively unique to individuals, an individual's ERP pattern, or "brainprint", can be used for personal identification in a manner analogous to fingerprints. Second, because there is remarkable degree of stability in individual waveforms over time under identical recording conditions, it may be possible to identify critical changes in individual ERP patterns which can be used to assess job performance and functional impairment due to fatigue, stress, alcohol and drug abuse, and other factors. Other potential uses of the individual identification system include security/intelligence/interrogation, personnel reliability identification and assessment, neonatal and infant identification and assessment.

Neural Network Technology

One problem which has plagued the interpretation and use of bioelectric and biomagnetic data is the shear complexity of the brain networks which generate the data. There are numerous neural networks in the brain, and these networks have very complex interconnections, and nonlinear response patterns. Relationships between the latencies, and amplitudes of ERP and ERF waveform features are becoming increasingly well understood. In addition, there are many individual variations in waveform morphology which complicate the identification of specific waveform features. Recently, new computing techniques which are modeled after brain neural functioning have been developed. As a general class, these are called neural network analysis techniques. This neural network analysis technology offers a method for finding complex, nonlinear relationships in large data sets, even when the nature of the relationships is not known in advance. Neural network technology is most often implemented using computer software programs, but hardware implementations of the technology are also available. Neural network theory, and detailed descriptions of specific techniques are available in numerous books and articles (Dayhoff, 1990; Rumelhart & McClelland, 1986; McClelland & Rumelhart, 1986; Wasserman, 1989; Ryan-Jones & Lewis,

1992). The unique feature of this technology is the capability to learn which features of a data set can be used to classify the examples into either unknown or predetermined categories. There are a variety of neural network techniques which could be used to classify ERP or ERF patterns. Neural networks may differ in the way the elements are interconnected, the way the data are processed, as well as the way in which the network structure is modified during learning. In most networks, input data values are adjusted through a series of layers by a series of transforms and weights so that the output category is correctly predicted. For example, if all of the possible examples are contained in the data set, then a self-organizing network could be used to classify the brainwave data. If only some of the possible examples are in the data set, then a network which utilized supervised learning could be more appropriate. The most commonly used, and best described network is the backpropagation network. This network is named because the error in output classification during training is used to adjust the weights at each level in the network in a backward fashion.

METHOD

Commercially available electrodes, made of tin to minimize depolarization, were attached to the scalp and conformed to the location standards of the 10/20 International System (Jasper, 1958). The electrodes were attached at the parietal (PZ) and frontal (FZ) sites, and referenced to the left mastoid region (A1). Additional electrical voltage was recorded from A1 referenced to the right mastoid area (A2). The electrical activity from A1 and A2 were averaged, in a common technique called digital re-referencing. The electrical voltage picked up by the electrodes was very small (microvolts, millionths of a volt) and amplified and filtered. To ensure adequate recording attachment, the impedance was measured prior to recording. Meter readings were 5 KOhms or less. Amplifier gain was 20000 times and the filter bandpass setting was 0.1 - 100 Hz. The amplified signals were then fed to a computer-based data acquisition system. Sampling rate was 128 Hz.

The event-related brain potential, ERP, was processed by removing unwanted artifacts such as eye blink or muscle movement, and specific single epochs were selected. The ERP data were "windowed" in order to reduce the number of inputs to the neural network. Windowing refers to taking a specific number of points along the ERP waveform, such as 6, and averaging them together.

Commercial software was used to create, train, and test the neural network. The neural network analysis package included software to convert the data file into a definition file and a fact file. The definition file provided the specifications on how the network was to be built, the number of input, hidden and output neurons, the data type, and information about screen display. The fact file specified the input and training pair pattern information. Training of the network was done using a backpropagation learning algorithm, which produced trained network files. After training of the network was complete, the neural network was tested using new data.

A 3-layer backpropagation network may be sufficient if the data set is relatively small (e.g., less than 100 individuals in the data base). The network may only require a single hidden layer with the same number of processing elements as inputs. One important step is the training, and testing of the network. Half of the ERPs from the sample were used to train the neural network to distinguish between each individual. The other half of the ERP were used to verify that the neural network was performing at the required level. Once the neural network had been trained, the network was used to make decisions about new samples of ERPs. This process is shown schematically in Figure 1. Samples of ERP data from four (4) individuals, named John, Greg, Jim and David are shown as being processed by the neural network. Identification of the individuals may be seen at the output layer.

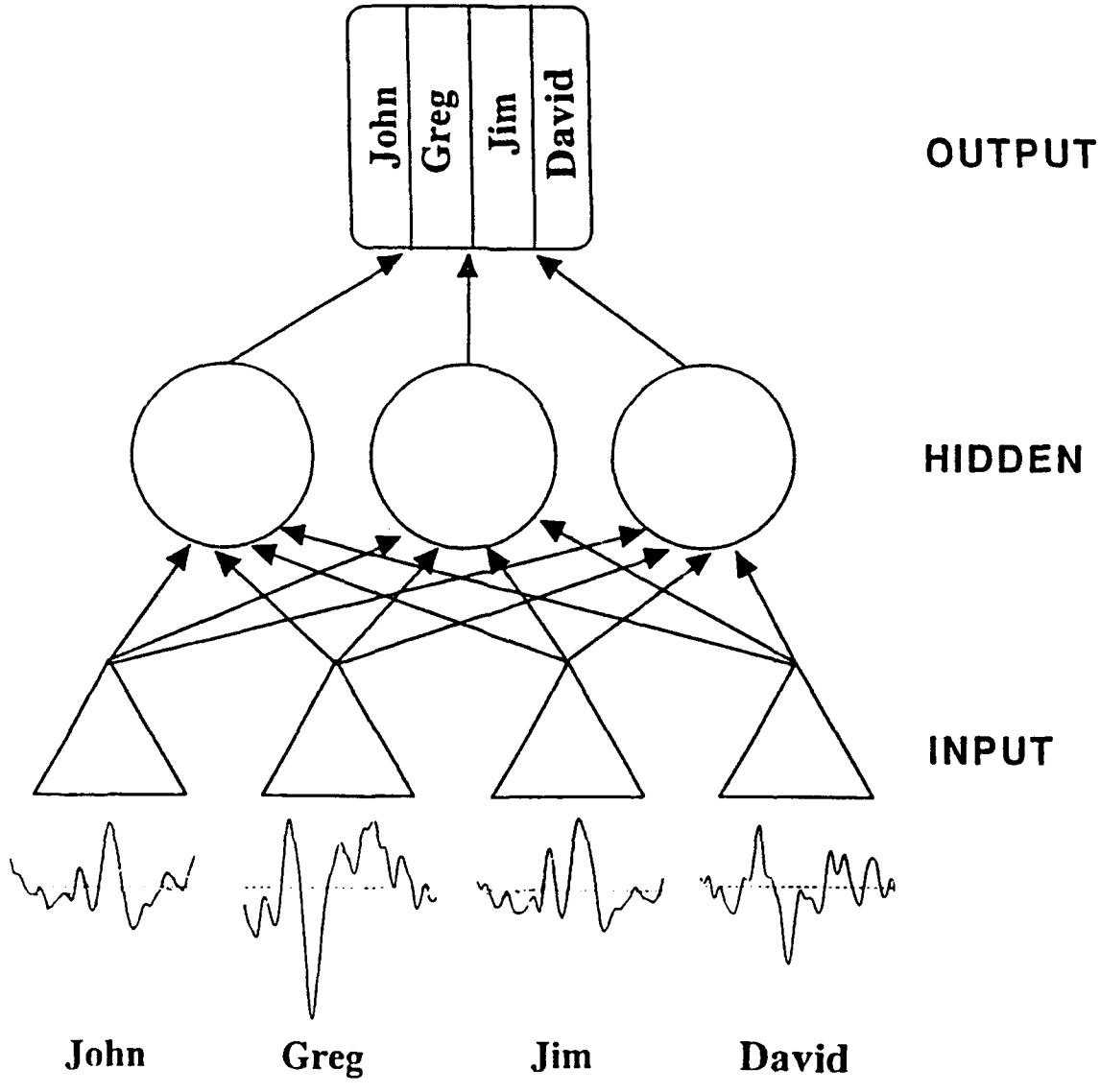


Figure 1. Schematic diagram of ERP identification by neural network analysis

The subjects for the preliminary test were 35 male Marine Corps personnel. Each Marine performed 400 trials of a two-letter ("o" and "c") discrimination task. ERP data were recorded from site PZ referenced to the digitally-linked mastoid regions. Site PZ is located over the parietal region of the brain, which is believed to be a sensory association/integration area. The ERPs were sampled from -200 ms prestimulus to 1000 ms poststimulus at 128 Hz, 20000 amplifier gain, and 0.1-100 Hz filter bandpass. The ERP data were divided into 8 blocks of 50 trials (400 total). The first ten (10) trials in each block with correct behavioral responses (hits) were averaged by conventional methods to obtain 8 ERPs for each of the 35 subjects (280 ERPs total). To reduce the number of input variables, each ERP was divided into 25 windows. These windows were about 47 ms wide and consisted of 6 data points each. The mean of these 6 data points was obtained for each window. The result was that 25 variables for each ERP were input to the neural network instead of 128 variables (128 Hz sampling rate translates to 128 variables per 1 second sample). A backpropagation network was used for training the network and develop the classification algorithm for the ERP data. The 3-layer network consisted of an input layer with 25 elements (ERP windows), a hidden layer with 25 elements, and an output layer with 35 elements (individual subjects).

RESULTS

The ERPs for each subject were divided into training and test sets. The training set consisted of the ERP data from the odd-numbered blocks, and the test set consisted of the ERP data from the even-numbered blocks. All of the examples in the training set (4 ERPs for each of the 35 subjects) were correctly learned to the required criterion. The neural network was then tested using the different ERPs from the test set. The network correctly classified 70/140 (50%) of the ERPs based on the highest output value. These results were statistically significant given that each of the 35 subjects was a separate output category. An additional metric was used to evaluate the correctness of classification for the subjects. For testing, the network needed to correctly classify 2 of the 4 ERP samples for each subject. Using this metric, the network correctly classified 29/35 (83%) of the ERPs.

Findings from the above preliminary neural network analysis were replicated and extended. ERP data from 40 male Marine Corps personnel were used. Subjects were not preselected on any factor, including task or job performance. Data from an additional recording site over the frontal region of the brain (FZ) was added to the data from PZ to allow for more classification features. Recording and averaging of the ERP records were the same as reported above. However, the 5 prestimulus windows for each site were deleted from the data set. As above, the 400 trials were divided into 50 trials per block, yielding 8 ERPs per subject. Again, only the first 10 trials were used to generate each ERP. The study used 320 ERPs total (8 ERPs/subject X 40 subjects = 320 ERPs). There were 160 ERPs in each of the training and testing sets. The 3-layer neural network consisted of an input layer of 40 elements (20 ERP windows from each site), a hidden layer with 40 elements, and an output layer with 40 elements (subjects). All of the ERPs were correctly classified during training, and a substantial improvement in the classification of the test examples was seen during testing. The network correctly classified 117/160 (73%) for ERPs in the test set. Using the

classification metric described above, of at least 2 of 4 ERPs for each subject, 39/40 (97.5%) of the subjects were correctly classified.

DISCUSSION

Even though the current paper deals with neuroelectric ERP recordings, similar equipment and procedures may be used in the recording, processing, and analyzing of neuromagnetic evoked field (EF) data. Descriptions of hardware and software recording equipment and procedures have been published elsewhere (Lewis, 1983; Lewis, Blackburn, Naitoh, & Metcalfe, 1985; Lewis, Trejo, Nunez, Weinberg, & Naitoh, 1987; Lewis & Sorenson, 1989; Lewis, Trejo, Naitoh, Blankenship, & Inlow, 1989). These publications reported on data using a single channel neuromagnetometer, however, NPRDC has a 5 channel neuromagnetometer system for the recording of EF data over more channels and larger number of brain regions. Even though the data reported in this paper are restricted to ERP data, earlier research has suggested that EF data may provide improved identification due to being non-contact in nature, monopolar, providing improved localization of brain activity, and providing improved sensitivity to individual subject differences (Lewis, 1983).

There are no other known ways to record brain function, in a practical way, other than neuroelectric contact electrodes or neuromagnetic pickup sensors. The latter sensors need not be in contact with the scalp directly to sense the biomagnetic activity from individuals. Positron emission tomography (PET) technology is able to describe anatomical relationships, and some physiological processing. However, PET is very expensive and does not have adequate temporal resolution for effective assessment of cognitive processing. Several minutes of data recording are required to show brain processing. The temporal resolution, required to assess dynamic cognitive processing, is improving with PET, but still lacks the millisecond resolution found with ERP recordings. PET is also an "active" technology requiring the injection of labeled radioisotopes to function. The described ERP/ERF technology is totally "passive," in that no energy or material is needed to obtain the ERP/ERF data. Alternative technologies such as computerized axial tomography and magnetic resonance imaging are possible candidates for personnel identification, but are extremely expensive, are "active" systems, and suffer from the same limitations as the other anatomically-based systems (e.g., fingerprint) noted above.

Traditional statistical techniques are an alternative to neural network analysis. However many assumptions must be made of the data, and these techniques are insensitive to nonlinear processes. Neural network analysis techniques do not make *a priori* assumptions about the input data, and are sensitive to nonlinear characteristics, which are found in biological recordings. As a result, neural network analyses have the potential to provide greater accuracy in the classification of complex and nonlinear data, such as found in brain recordings, than the traditional statistical techniques.

REFERENCES

Dayhoff, J. (1990). *Neural network architectures: An introduction.* New York: Van Nostrand Reinhold.

Jasper, H. (1958). The ten-twenty electrode system of the International Federation. *Electroencephalography and Clinical Neurophysiology, 10,* 371-375.

Lewis, G. W. (1983). Event related brain electrical and magnetic activity: Toward predicting on-job performance. *International Journal of Neuroscience, 18,* 159-182.

Lewis, G. W. (1984). Temporal stability of multichannel, multimodal ERP recordings. *International Journal of Neuroscience, 25,* 131-144.

Lewis, G. W. & Sorenson, R. C. (1989). Evoked brain activity and personnel performance. In Dillon, R. F. & Pellegrino, J. W. *Testing, Theoretical and Applied Perspectives.* New York: Praeger.

Lewis, G. W. & Ryan-Jones, D. L. (1992). Neural network identification of individuals from ERP patterns. Presented at the Fourth Annual Convention of the American Psychological Society, San Diego, 21 June 1992.

Lewis, G., Blackburn, M., Naitoh, P., & Metcalfe, M. (1985). Few-trial evoked field stability using the DC SQUID. In Weinberg, H., Stroink, G., & Katila, T. *Biomagnetism: Applications and Theory.* New York: Pergamon Press.

Lewis, G. W., Trejo, L. J., Nunez, P., Weinberg, H., & Naitoh, P. (1987). Evoked neuromagnetic fields: Implications for indexing performance. In Atsumi, K., Kotani, M., Ueno, S., Katila, T., & Williamson, S. J. *Biomagnetism '87.* Tokyo: Tokyo Denki University Press.

Lewis, G. W., Trejo. L. J., Naitoh, P., Blankenship, M., & Inlow, M. (1989). Temporal variability of the neuromagnetic evoked field: Implications for human performance assessment. In Williamson, S. J., Hoke, M., Stroink, G., & Kotani, M. *Advances in Biomagnetism.* New York: Plenum Press.

McClelland, J. L. & Rumelhart, D. E. (1986). *Parallel distributed processing. Volume 2: Psychological and biological models.* Cambridge: The MIT Press.

Naitoh, P. & Lewis, G. W. (1981). Statistical analysis of extracted features. In Yamaguchi, N. & Fujisawa, K. (Eds.). *Recent advances in EEG and EMG processing.* Amsterdam: Elsevier/North Holland Biomedical Press, pp 179-194.

Rumelhart, D. E. & McClelland, J. L. (1986). *Parallel distributed processing. Volume 1: Foundations.* Cambridge: The MIT Press.

Ryan-Jones, D. L. & Lewis. G. W. (1992). Neural network analysis of event-related potential (ERP) data. In: Montague, W. E. (Ed.). *Independent Research and Independent Exploratory Development Programs: FY91 Annual Report.* (NPRDC Report AP-92-5). San Diego: Navy Personnel Research and Development Center.

Wasserman, P. (1989). *Neural computing: Theory and Practice.* New York: Van Nostrand Reinhold.

Category Learning in a Hidden Pattern-Unit Network Model

Joshua B. Hurwitz

Armstrong Laboratory

Learning Abilities Measurement Program

The major goal in research on individual differences has been to identify cognitive abilities underlying tasks that humans perform. In the traditional factor analytic method, abilities are deduced by analyzing the variance shared by a set of standardized tests (e.g. Tirre & Pena, in press). However, in the approach taken here, abilities are built into process models, rather than being inferred from factor analysis. Such models are used to generate predictions of performance on a simple cognitive task. Individual differences can be indexed by comparing how well models with differing mechanisms fit data from various kinds of subjects. Also, such differences can be measured using the estimated values of free parameters in a model.

Another distinction between the factor-analytic and process-model approaches is in the type of data collected. Rather than analyze results from several standardized tests, as in the factor-analytic method, a process model is tested on an artificial task that is related to many human activities. The task employed here, category learning, relates to most types of processing, including identification and recognition (Nosofsky, 1988). In fact, it is difficult to conceive of a process that does not involve classifying a stimulus.

In artificial category learning, a subject is given a series of trials in which they are trained to classify a set of stimuli. On each trial, a stimulus is presented and the subject is asked to respond with a category label. After the response is made, the subject is told the correct category.

One aim in modeling category-learning data is to predict trial-by-trial categorization probabilities from a group of subjects. In order to accomplish this, the model and all subjects in the group must be trained using the same trial sequence. This provides a more rigorous test of the model than presenting a different random sequence to each subject and having the model predict averages over blocks of trials. Being required to predict trial-by-trial performance forces the model to take both category structure and training history into account when it generates an acquisition curve.

The model presented here, the Hidden Pattern-Unit Network model (HPU), incorporates three mechanisms that are currently being studied in category learning: pattern storage and retrieval, non-linear similarity (Estes, Campbell, Hatsopoulos & Hurwitz, 1989; Kruschke, 1992; Medin & Shaffer, 1978; Nosofsky, 1988), and error-correction learning (Gluck & Bower, 1988). HPU assumes three levels of representation: feature nodes, hidden pattern-units and category nodes. Each feature node has a weighted connection to all hidden pattern-units, and every hidden unit has a weighted connection to each category node. The hidden pattern-units store previously presented training patterns, and a new hidden unit is created each time a novel training pattern is presented.

When presented with an input pattern, the model activates the feature nodes, hidden units and category nodes, and then computes categorization probabilities. For

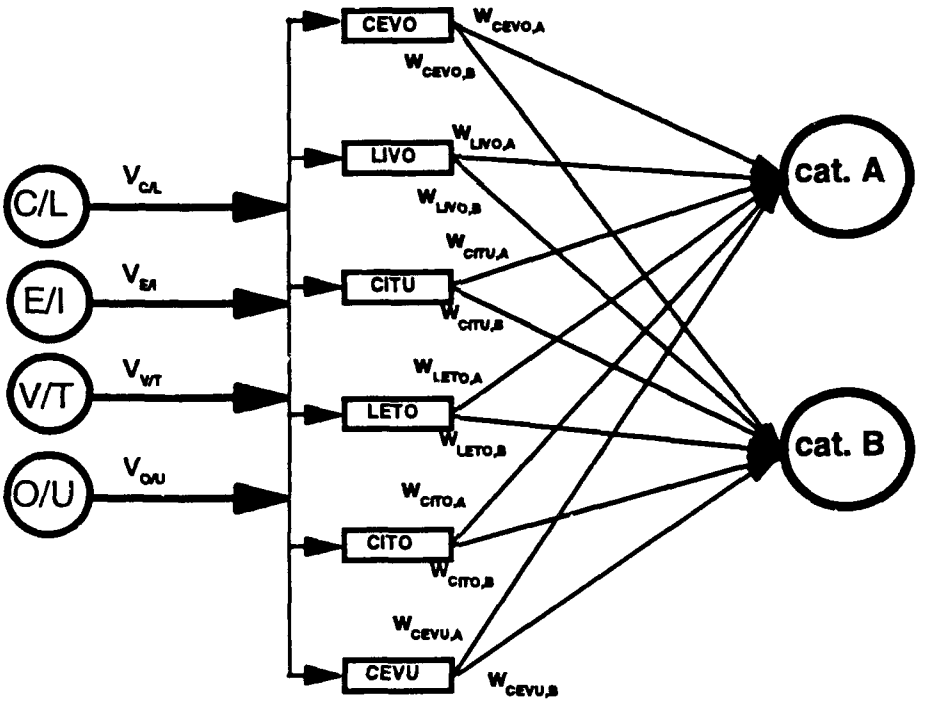


Figure 1. The Hidden Pattern-Unit Network Model.

---

the example shown in Figure 1, each feature node detects which one of two features is present in the input. In this example, the input patterns are 4-letter non-words, and there can be one of two possible letters present in each position of a non-word. One letter produces an activation of 1 on its corresponding feature node and the other produces and activation of 0.

The model activates a hidden unit based on the similarity between the input pattern and the pattern stored at that unit. The activation of hidden unit h is

$$a_h = \prod_i s_i^{\,|a_i - a_{ih}|} \tag{1}$$

where a_i is the activation of feature node i, a_{ih} is the value for the corresponding feature stored at hidden unit h, and s_i is the similarity parameter for node i. The similarity parameter is a logistic function of its corresponding feature weight, so that

$$s_i = \frac{1}{1 + e^{-[v_i + c_T]}} \tag{2}$$

where v_i is the weight on feature-node i and c_T is a free parameter ($-\infty < c_T < \infty$).

After activating hidden units, the model computes outputs to the category nodes and transforms the outputs to categorization probabilities. At the category level, the output function is the one used by Gluck & Bower (1988), and the probability function is the one used in Estes et al. (1989). The output for category node k is

$$o_k = \sum_h a_h w_{hk} \tag{3}$$

where w_{hk} is the weight connecting hidden-unit h and category-node k. When presented with input pattern $\mathbf{x}$, the probability of giving category C_k as the response is

$$P_x(C_k) = \frac{e^{co_k}}{\sum_j e^{co_j}} \tag{4}$$

where the sum is over all category nodes and c is a free parameter ($c > 0$).

After computing categorization probabilities, the model learns based on feedback from a teaching signal. For learning at the category level, the model uses the delta-rule function (Gluck & Bower, 1988):

$$w_{hk,t+1} = w_{hkt} + (z_k - o_k)\, a_h\, \beta_H \tag{5}$$

where t is the trial number; β_H, the hidden-to-category learning rate, is a free parameter ($0 \le \beta_H \le 1$); and z_k, the teaching signal, is 1 when category C_k is the correct category, and 0 otherwise. For learning at the feature level, the model uses back propagation, so that

$$v_{i,\,t+1} = v_{it} + (1 - s_i)\beta_F \sum_h |a_i - a_{ih}|\, d_h a_h \tag{6}$$

where β_F, the feature-unit learning rate, is a free parameter ($\beta_F > 0$), and d_h, the propagated error, is

$$d_h = \sum_j (z_j - o_j) w_{hj} \,. \tag{7}$$

In testing HPU, the focus was on analyzing the feature-learning mechanism shown in Equation 6. As a comparison model, a non-feature-learning version was developed in which there is one feature weight, v, for all features. In this case

$$a_h = \prod_i s^{|a_i - a_{ih}|} \tag{8}$$

and

$$v_{t+1} = v_t + (1 - s)\beta_F \Big\{ \sum_h d_h a_h \Big[\sum_i |a_i - a_{ih}| \Big] \Big\} \tag{9}$$

where

$$s = \frac{1}{1 + e^{-[v + cT]}} \,. \tag{10}$$

Both models, HPU with and without feature learning, were fitted to data from a 320-trial category-learning task. On each trial, a subject was presented with a non-word, was asked to classify it into category A or B, and, after responding, was given the correct category. Non-words beginning with the letters CE or LI always belonged in category A, whereas those beginning with CI or LE were always in category B. Thus, the first two positions of each non-word were "globally relevant" for categorizing.

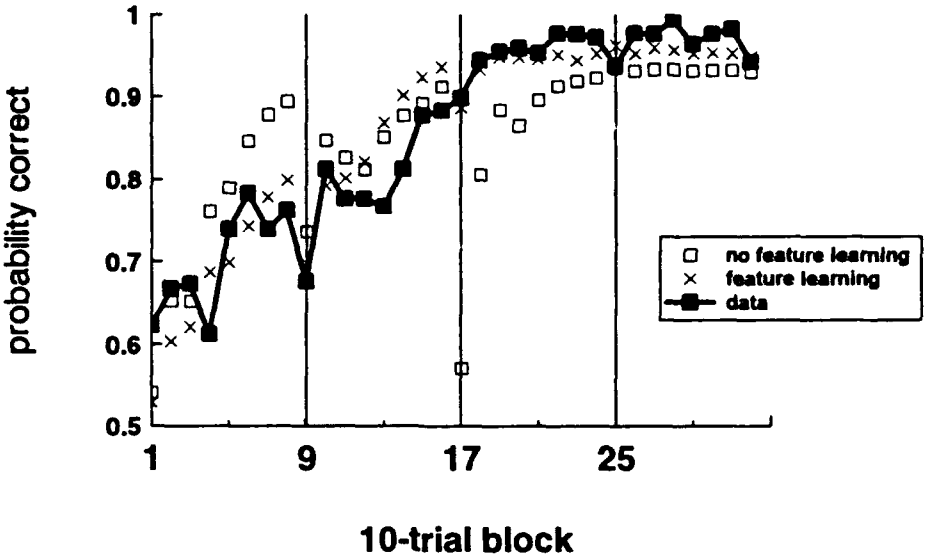


Figure 2. Acquisition curves for the solvers, and model fits for the two versions of HPU. Each data point is an average over all solvers for a block of 10 trials. The trial divisions are demarcated by the vertical lines.

The last two positions were occasionally relevant. Non-words ending in VU or TO were in category A for the first division of 80 trials, and were in category B for the third such division. However, for the remaining trials, these non-words occurred equally often in each category. Non-words ending in VO or TU followed this same sequence, except that they were in category B for division 1 and category A for division 3.

Using these criteria, a single sequence of training trials was generated, and all subjects were trained with this sequence. However, the positions that were globally relevant were varied from subject to subject to prevent positional biases from influencing the results

For the results, one focus was on the requirements for producing optimal performance on this task. If, by the end of the second division of 80 trials, the learner was using only globally relevant letters to categorize the non-words, then performance would be perfect throughout the remainder of training. However, if the learner was using at least one other letter, then performance would falter in division 3.

Figures 2 and 3 show the data for two groups of subjects, those who gave the correct categorization rule at the end of training (solvers, N=18), and those who did not (non-solvers, N=16). The solvers' performance climbed steadily early in training, and asymptotically exceeded 90% (Figure 2). The non-solvers' performance dropped at the beginning of division 2, and rose in a zigzag fashion in the next two divisions (Figure 3). It also dropped in division 4, even though the subjects had been previously presented with all 16 non-words.

As Table 1 and Figure 2 show, the feature-learning model was better than the no-feature-learning model at fitting the solvers' data. In particular, the feature-learning model was better at predicting the solvers' ability to generalize at the beginning of division 3. The other model showed a decrement in performance in this part of the training sequence.

Model	Solvers		Non-Solvers
	Overall	Beginning of Div. 3 (Blk. 17)	Overall
Feature learning	0.11	0.09	0.18
No feature learning	0.14	0.28	0.18

Table 1. The trial-by-trial standard errors of the model fits for HPU.

The reason HPU predicted solvers' performance better with feature learning than without it can be seen by assessing the value of s_i at the end of division 2. The lower the value of s_i, the more the model is "attending" to feature i. In the feature-learning model, the average s_i at the end of division 2 was 0.15 for the two globally relevant letters and was 0.51 for the remaining two features. Thus, the use of feature learning in HPU allowed the model to use only the globally relevant features and to ignore the features that had been irrelevant right before division 3.

Although the addition of feature learning improved the fit of HPU to the data, neither the feature-learning nor the no-feature-learning model gave as good a fit to the non-solvers' data as it had to the solvers' data (Table 1). Both versions of the model produced similar acquisition curves, and these curves were too smooth compared to the data. For example, the data curve showed a sharp drop at the beginning of division 4, whereas the model curves showed a continuous increase from division 3 through division 4 (Figure 3).

One possible reason for this lack of fit could have been that the solvers and non-solvers were not using the same learning rule. The evidence for this was that a "non-interactive" learning rule provided a better fit than the delta rule to the non-solvers' data (Hurwitz, 1990). Unlike the delta rule, which, like multiple regression, estimates weights by minimizing an error, the non-interactive rule minimizes a different error for

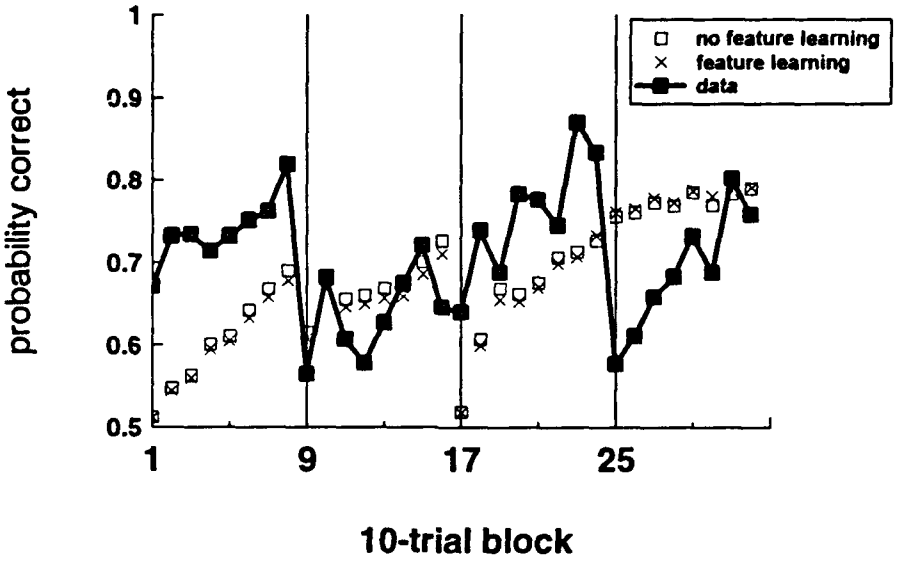


Figure 3. Acquisition curves for the non-solvers, and model fits for the two versions of HPU. The trial divisions are demarcated by the vertical lines.

each hidden unit. This produces a form of interference whereby associations built up early in training are eliminated by later training. This interference led the model to predict a drop in performance at the beginning of division 4.

Though the use of feature learning in HPU appeared to improve the model's fits to the solvers' data, this model has problems as well. The use of back propagation can lead the model to alter its feature weights, even when subjects do not appear to alter which features they consider important. For example, consider the above experiment, except that the categories are reversed in division 3. Thus, in division 3, non-words beginning with CE or LI are in category B, those beginning with CI or LE are in category A, those ending with VU or TO are in category A, and those ending with VO or TU are in category B. When presented with such a shift, a solver could either continue to use the globally relevant positions (1 and 2 in this example), and just alter category associations, or could switch to other positions. When a reversal was implemented in a second experiment, solvers behaved as if they were doing the former, and the HPU feature-learning model did the latter.

The effect of reversing the categories could be seen by looking at results from test trials presented periodically during training. These trials were just like the training trials, except that no feedback was provided. Also, the non-words used on test trials consisted of letter pairs from different categories. For example, in division 3, one test pattern, CEVU, would be an *aabb* non-word, in that the first two letters were presented in category A during division 3, and the other letters were presented in category B.

Post-reversal test trials were used to identify which letters subjects employed to categorize the non-words, and pre-reversal test trials were used to identify subjects who had successfully solved the problem. A solver was defined as a subject who had achieved perfect performance on 4 test trials immediately preceding the reversal, and who had achieved at least 80% correct on the last 16 training trials preceding the reversal.

Figure 4 shows that in division 3, after the reversal, the probability of making a category-*A* response given an *aabb* test pattern, $P(C_A/aabb)$, was higher for the solvers than for the model, and $P(C_A/bbaa)$ was lower for the solvers than for the model. The reason for this was that after the reversal, feature nodes for the globally relevant positions of each non-word had a higher average value for s_i (0.65) than did nodes for the other positions (0.07). The opposite had occurred before category reversal, with average s_i being lower for nodes in the globally relevant positions (0.06) than for those in the other positions (0.54). When the reversal occurred, negative error flooded the network at the feature level, leading to a reversal of the feature weights. Thus, category reversal changed which letters the model considered "important," whereas it apparently had no such effect on the subjects.

This result was replicated in another condition in which subjects were presented with figures instead of non-words. The figures varied in shape (square versus triangle), size (large versus small), number (1 versus 2) and position (top of computer screen versus bottom). As with the non-word stimuli, the solvers and the model appeared to "attend" to different features after the reversal (Figure 4). Also, as with the non-words, the model's average s_i after the reversal was higher for globally relevant features (0.46) than for the remaining features (0.20).

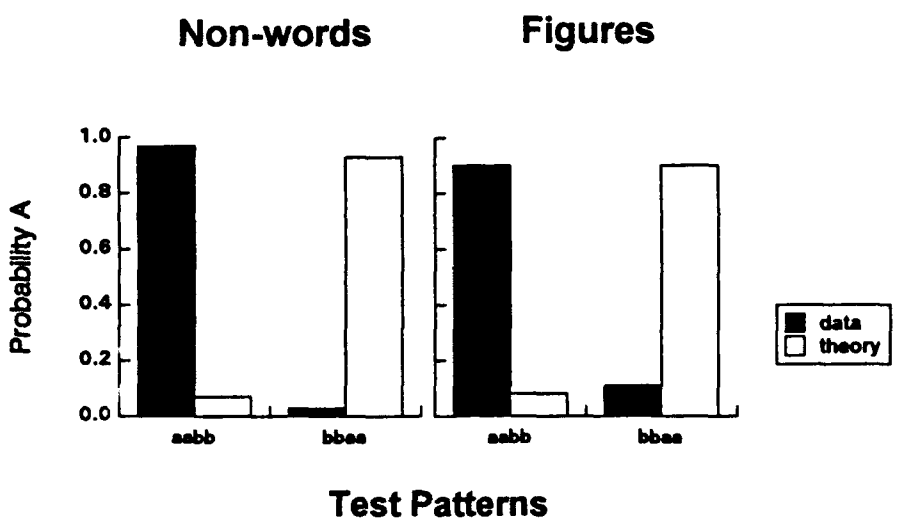


Figure 4. Division-3 (post-reversal) test-trial data and predictions for reversal study. The data are from subjects who had solved the problem before the reversal. (N=37 in the non-word condition, and N=60 in the figure condition.)

The outcomes of these studies demonstrate the usefulness of the process-model approach for identifying cognitive abilities, and for indexing individual differences. One important result was that, among all of the models, HPU with feature learning was the best at predicting the solvers' performance in the first experiment. Without feature learning, the model could not maintain a high level of performance in the third division of training trials.

Even though HPU with feature learning accurately predicted the solvers' acquisition curve, it was not as good at making other predictions. First, it produced inferior fits to the the non-solvers' curve. One reason for this may have been that the solvers and non-solvers were using different learning mechanisms. However, the model even fell short in predicting the solvers' performance in the second experiment. When confronted with a category reversal, HPU incorrectly predicted that the solvers would change which features are important for categorizing.

These results show that given hidden units that store patterns, a back propagation model that learns which features are relevant can provide accurate predictions of training performance. However, such a model does not always adapt as humans do to changes in category structure. Perhaps the appropriate model would be one in which the feature learning rate (β_F) decreases to near 0 once asymptotic performance has been achieved. Such a model might account for the fact that subjects do not appear to alter which features they consider important when categories are reversed. This dynamic-learning-rate assumption is currently being explored.

References

Estes, W. K., Campbell, J. Hatsopoulos, N., & Hurwitz, J. B. (1989). Base-rate effects in category learning: A comparison of parallel network and memory storage-retrieval models for category learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 15*, 556-571.

Gluck, M. A., & Bower, G. H. (1988). From conditioning to category learning: An adaptive network model. *Journal of Experimental Psychology: General, 117*, 227-247.

Hurwitz, J. B. (1990). *A hidden-pattern unit network model of category learning*. Unpublished doctoral dissertation, Harvard University, Cambridge, Massachusetts.

Kruschke, J. K. (1992). ALCOVE: An exemplar-based connectionist model of category learning. *Psychological Review*.

Medin, D. L. & Shaffer, M. M. (1978). Context theory of classification learning. *Psychological Review, 85*, 207-238.

Nosofsky, R. M. (1988). Exemplar-based accounts of relations between classification, recognition, and typicality. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 14*, 700-708.

Tirre, W. C., & Pena, C. M. (in press). Components of quantitative reasoning: General and group ability factors. *Intelligence*.

Causal Structure, Neural Networks, and Classification[1]

Christopher Meek
Department of Philosophy,
Carnegie Mellon University

Richard Scheines
Department of Philosophy,
Carnegie Mellon University

§0 Motivation

Neural network technology has been successfully applied to recognition and classification problems. For example, neural networks have been used to recognize hand written letters and digits. A neural network has a structural component and a quantitative component. Techniques for both "structure learning" and "weight learning" have been developed for neural networks, although the second topic is much more developed [Ash 89][Frean 90][Karnin 90]. The structure of these networks are thought by many to be models of organic neural structures. Despite the possible connections to organic neural structures in application to recognition and classification, the structure of these networks bears no clear relation to the causal structure governing the variables involved. Such networks cannot be used (in any obvious way) to predict the effect of interventions or policy changes.

Bayesian networks form another related technology used in the artificial intelligence and statistics communities. Bayesian networks are useful in classification and prediction problems. Some of the advantages of using Bayesian networks are the following; There is no need to predetermine the inputs and outputs, that is, a Bayesian network can be used as a classifier for any set of variables in the network and the inputs for that classification can be any subset of the remaining variables in the network. This feature is useful in making classifications where only partial information is available. In addition, the time to parameterize a network -- this process is similar to training a neural network -- is linear in the size of the data set. Given a Bayesian network that correctly represents causal connections among variables one can make qualitative and quantitative predictions on the effects of an intervention or policy change[Spirtes et al 92]. Perhaps most importantly, given some weak assumptions, there are provably reliable algorithms for constructing the network structure.

This paper considers two questions;

1) What are the relations between Bayesian networks and neural networks? We show that a broad class of neural networks are in fact Bayesian networks.

2) Is it important for classification and recognition problems to use a network that does not misrepresent the causal relations among variables? We will show that the correct representation of the causal structure does matter even for recognition and classification problems.

---

[1] Research for this work was funded by the Navy Personnel Research and Development Center and the Office of Naval Research under grants #N00014-88-K-0194, N00014-89-J-1964 and #N00014-91-J-1361. We thank Clark Glymour and Peter Spirtes for comments and suggestions on this paper.

§1 The connection between Neural networks and Bayesian networks

§1.1 Neural networks

Neural networks have become popular in the artificial intelligence community principally for two reasons: one, they are thought by many to be interpretable as models of organic neural structures, and two, they have been made to "learn" a variety interesting and difficult classification tasks. Although the specifics can vary, neural networks have roughly the following general form. (figure 1).[2]

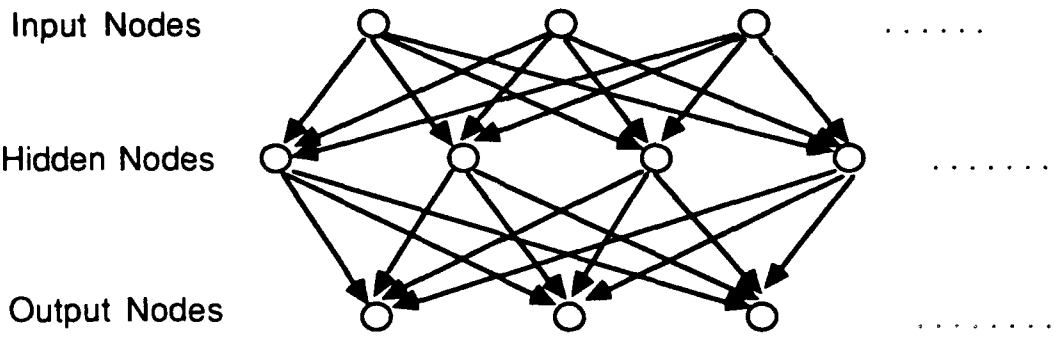


Figure 1

There is a layer of input nodes, a layer or layers of "hidden nodes," and a layer of output nodes. The connections between the nodes are usually directed, although they need not be. Each node has a certain "activation level," and each connection between nodes has an associated "weight." The activation level of non input nodes is some function of the nodes that feed into it. This function may have a stochastic component. The general form of the function is usually fixed ahead of time, and the class of functions that determine a particular node's activation level is then parameterized by the weights. For example, suppose node y_i is connected to nodes x_1-x_j, where the weight associated with a node from x_j to y_i is written $w_{i,j}$. We might specify that the class of functions connecting y_i to its inputs is linear:

$$y_i = \sum_{k=1}^{j} w_{i,k} x_k$$

Alternatively, we might specify that it is quadratic:

$$y_i = \sum_{k=1}^{j} w_{i,k}\left(x_k^2 + x_k\right)$$

or we could specify some other function.

The network "learns" by changing the weights in order to improve its predictive ability.[3] Neural network learning is therefore a form of parameter estimation, and is explicitly treated as such by many researchers [Nowlan 91].

---

[2] See the text by [Feeman and Skapura 91], for example.

[3] Much of the research in the field today concerns particular learning schemes and their properties.

If the network were only two layers, one for inputs and one for outputs, then the problem of describing the general properties of the network and its training regime would be fairly simple. The class of functions from inputs to outputs it could possibly learn would be given by the pre-specified class of functions connecting the outputs to the inputs, and the properties of various learning schemes, or estimators, would be more or less understood. If there are "hidden nodes" and the network is more than two layers, however, the overall function connecting the inputs to the outputs is really a composition of the functions connecting each layer inbetween. Not all functional classes are closed under composition, and in some cases the class of functions computable by a network is not known. The same analysis holds if one considers the network as stochastic. A distribution can be given for the input nodes, and the non input nodes can be pre-specified to have some class of conditional distributions on nodes feeding into them, parameterized by the weights. But not all families of distributions are convex, and thus a network of more than two layers might be capable of learning a space of joint distributions quite different from the family specified between layers.

Nevertheless, by specifying the topology of the network and the family of functions connecting nodes to their inputs, one picks out some region in the space of all possible functions from inputs to outputs (figure 2).

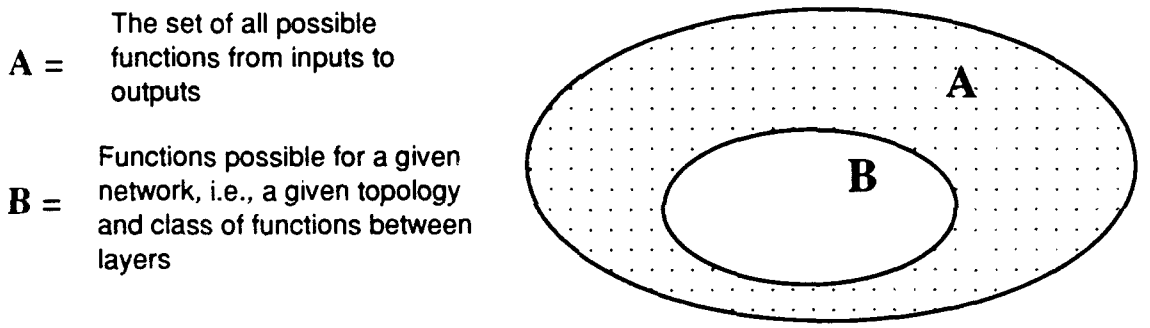


Figure 2

Learning the weights amounts to trying to find the point in this region that minimizes some loss function or that maximizes some likelihood function. The weights are given a default value, and then incrementally modified according to any of a variety of schemes, e.g. back propagation, annealing, etc.

Network topology is important in that it defines the class of functions that are learnable. If one is wrong about the region of functions to be estimated, then no learning procedure can help (figure 3).

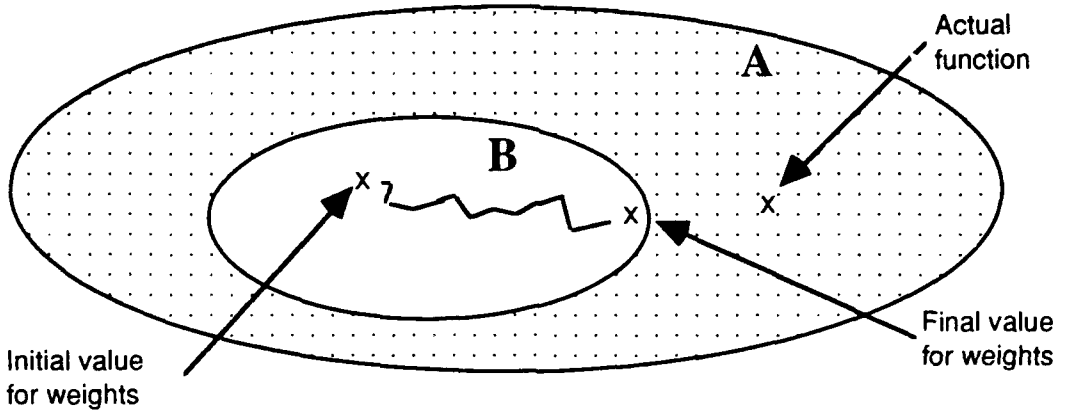


Figure 3

§1.2 Bayesian networks

A Bayesian network is usually defined to be a tuple <G,P> where G=<V,E> is a directed acyclic graph over the vertices V and with the edges E. P is the probability distribution over the random variables which correspond to the vertices of the graph G. Each directed acyclic graph G=<V,E> represents a set of probability distributions, each of which can be factored according to the following rule,

$$P(V) = \prod_{x \in V} P(x \mid \pi_x)$$

where π_x is the set of parents of x in G. Consider the following example.

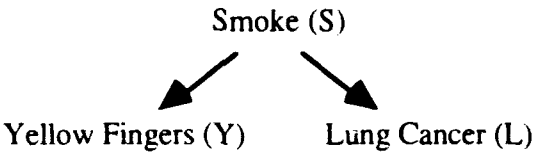


Figure 4

The joint probability distribution over the variables S, Y and L factor into the following equation. P(S,Y,L) = P(S) P(Y|S) P(L|S). Let each of the variables be Boolean variables where S takes on the values s or ~s, L can take the values l and ~l and Y can take the values y and ~y. One possible joint distribution is given below.

P(s) = 0.25
P(~s) = 0.75

P(y|s) = 0.7	P(~y|s) = 0.3	P(l|s) = 0.05	P(~l|s) = 0.95
P(y|~s) = 0.1	P(~y|~s) = 0.9	P(l|~s) = 0.16	P(~l|~s) = 0.84

As mentioned earlier, Bayesian networks are useful for predicting the effects of a manipulation. One might wish to asses the effect of manipulating the value of yellow fingers (Y) for members of a given population on the incidence of lung cancer (L). For instance, one might institute a policy of finger bleaching for all smokers. Given the Bayesian network in (figure 4) it is clear that the manipulation would have no effect of the prevalence of lung cancer. For a more detailed discussion of predicting the effects of manipulations see [Spirtes et al 1992] and [Spirtes et al 1993].

TETRAD II's Build procedure constructs Bayesian networks from data and any prior knowledge the user may have about the domain. We know that under weak assumptions the procedures in TETRAD II are asymptotically reliable, i.e., given the correct background assumptions, the probability that they will identify the correct equivalence class of topologies converges to one in the limit as the sample grows without bound.

§1.3 Neural networks and Bayesian networks

If the connections in a neural network are directed and there is no feedback, then the neural network *is* a Bayesian network. The difference is that in the Bayesian networks TETRAD II constructs there are no a priori constraints on the class of functions relating an

effect to its immediate causes. The only constraints on the region of functions the Bayesian network can compute is given by the topological -- i.e. graphical -- structure of the network, and these are to conditional independence constraints.

In a directed neural network without feedback the inputs precede the outputs (figure 1). But in using a Bayesian network, the inputs might be effects of each other and of the outputs. For example, were TETRAD II to construct a Bayesian network among variables x1, x2, y1, and y2, it might look as follows:

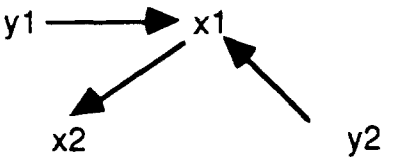


Figure 5

Despite the fact that the "outputs" precede the "inputs" one can use this (or any acyclic Bayesian network) Bayesian network to make predictions for any set of variables from any other set of variables with an inference process called "updating."

One of our research aims is to find ways that TETRAD II's Bayesian network builder or related algorithms can aid in specifying neural network topology, and do so in ways that are provably reliable. Converting Bayesian network topology constructed by TETRAD II to a neural network is often not straightforward and may not always be desirable. One problem is that in a neural network the inputs must precede the outputs, but there is no guarantee that this will be the case in any of the Bayesian networks constructed by TETRAD II.

The advantage of a neural network formalism is that once the weights are determined using the network to predict outputs from inputs requires very little computation. In Bayesian networks, by contrast, prediction by "updating" may require a large amounts of computation; a great deal of work has been done to improve updating speed using Monte Carlo methods.

A second fundamental problem is that in many cases the TETRAD II Build procedures indicate the possible presence of latent, unmeasured variables, and in such cases the algorithms do not produce a definite network. If the user has some prior knowledge as to which measured variables share a common, unmeasured variable, another procedure in TETRAD II, MIMbuild may in many cases be used to build a definite network.

§2 Bayesian networks as classifiers

Bayesian networks are commonly used as classifiers. In this section we will briefly describe one method of using Bayesian networks as classifiers. A training list, denoted by T, is a list of samples which is used to parameterize or "train" the Bayesian network. A sample point corresponds to one array of values for each of the variables (vertices). As mentioned above, each directed acyclic graph G=<V,E> represents a set of probability distributions, each of which can be factored according to the following rule,

$$P(V) = \prod_{x \in V} P(x \mid \pi_x)$$

where π_x is the set of parents of x in G. For a given training list T, the maximum likelihood estimate of the probability distribution over the variables V is obtained by substituting the

frequency of x given π_x in the training list T for $P(x \mid \pi_x)$[4] This is how the ESTIMATE procedure of TETRAD II parameterizes the network.[5]

Consider the following classification tasks. One might want to identifying which newly admitted heart attack patients are high risk patients and which are low risk patients.[6] Given the value of 19 variables that are measured during the first 24 hours we want to identify (classify) whether or not the patient will survive at least 30 days. The variable that we are interested in predicting or classifying is called the *target variable*, and will be denoted by TV where the target variable TV can take on the values t_1 to t_n. The inputs or evidence that we use to classify the target variable is called the *evidence set* which will be denoted by EV. The evidence set for the heart attack patient example would be measurements for the 19 variables.[7] The classification of the TV given EV is obtained by calculating the conditional probabilities P(TV | EV). The process of calculating the conditional probabilities is often called updating the network.

We can now describe how one can use a Bayesian network to classify the target variable. First update the network for the evidence set EV, then identify which values of TV are maximal for P(TV | EV). So the classification of TV given EV ($Class_P(TV,EV)$) is defined as follows,

$$Class_p(TV, EV) = \{t_i : \forall (1 \le j \le n) P(TV = t_i \mid EV) \ge P(TV = t_j \mid EV)\}$$

The classification of a target variable is a set of values which maximize the conditional probability of the target variable given the evidence set. One nice feature of using Bayesian networks for classification is that it possible to use a Bayesian network as a classifier for any subset of the variables in the graph. That is, the target variable need not be predetermined. Similarly, the evidence set can be any subset of the remaining variables.

§3 Structure matters

In §1.2 we noted that Bayesian networks can be used to predict the effect of interventions. The structure of the network is essential in calculating both qualitative and quantitative effects of an intervention or policy change.

For the network in (figure 6a) the intervention to stop smoking will have no effect on lung cancer. But if the network is as in (figure 6b) then an intervention to stop smoking will, other things remaining the same, affect lung cancer.

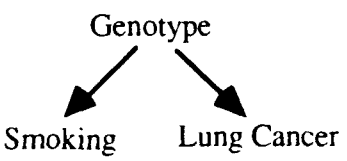


Figure 6a

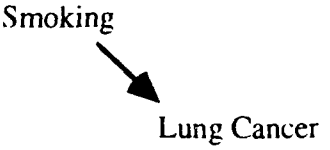


Figure 6b

---

[4] The maximum likelihood estimate of the parameters is well defined for positive distributions. For the results given in §3 we parameterized the Bayesian networks using the frequencies in the training list wherever possible. Where the frequency data fails to give a positive instance of some parent set we assume a uniform distribution over the values of the descendant of the parent set.

[5] See [Herskovitz 91] for a Bayesian estimation technique.

[6] This problem was studied by [Breiman et al 1984].

[7] Even if the measurements for all of the variables is not available it is still possible to classify the target variable. We simple define the evidence set to be the subset of the variables for which we know the values.

In this section we show that structure also matters for classification using Bayesian networks, that is, using the correct structure to classify lowers the expected number of classification errors.

§3.1 Measuring Error

Given the "true" Bayesian network that describes the relationship between the variables V in the graph we can measure the expected error of classification. Since there is a stochastic element involved one would certainly not expect there to be no classification error. One straight forward measure of the expected error is called Bayes error. The Bayes error for classifying the target variable TV from EV = V\{Y} where TV has n possible values, $t_1,..,t_n$ is BE(P,TV)[8] P is the joint distribution over the variables V of the true Bayesian network.

$$BE(P,TV) = \sum_{x \in EV} \left[\left(1 - \max_{1 \le i \le n} \{P(TV = t_i \mid X = x)\}\right) P(X = x) \right]$$

Intuitively, the Bayes error is the sum of the expected error for each of the possible instantiations of the evidence set EV with respect to the true joint distribution P(V). The expected error in classification for $x \in EV$ is simply 1 minus the conditional probability of the classification provided by the Bayesian network given the probability of that instantiation of the evidence.

We can generalize this formulation to compare the relative Bayes error (RBE) for a given probability distribution (P1) relative to the "true" distribution (P2). Again, TV is the target variable and EV = V\{Y} and P1 and P2 are distribution over V. Let $C_1(x)$ = $Class_{P1}(\{TV\},x)$ and c_1 to c_m be an enumeration of $C_1(x)$.

$$PE(P_1,P_2,TV,x) = \left[\left(1 - \sum_{1 \le i \le m} \frac{P_2(TV=c_i \mid X=x)}{m}\right) P_2(X=x) \right]$$

$$RBE(P_1,P_2,TV) = \sum_{x \in EV} PE(P_1,P_2,TV,x)$$

Note that $RBE(P_1,P_1,Y)$ reduces to $BE(P_1,Y)$ and that $RBE(P_1,P_2,Y) \ge BE(P_2,Y)$ for all P_1, P_2 and $Y \in V$. Again, more intuitively, the relative Bayes error is the sum over all instantiations in the evidence of the probable errors (PE) of the classification of TV on a particular instantiation $x \in EV$. The probable error is simply the probability that the classification from distribution P1 will be incorrect if the true distribution is in fact P2.

§3.2 Design of Experiment

The empirical data described in §3.3 below was obtained from the experiment described in this section. The experiment uses the Monte Carlo generator, the Estimate procedure, and the Makemodel module of TETRAD II. For all that follows, Y is the target variable and the evidence set EV = {X1,X2}. Each of the variables can take on three different values.

Experiment 1 - Let the true graph be G1 shown in (figure 7).

(1) We randomly parameterize the graph according the factorization described in §1.2 to obtain the "true" distribution P_t.

(2) From this distribution we can calculate the Bayes error, $BE(P_t,Y)$.

---

[8] A discussion of Bayes error can be found in [Breiman et al 84].

(3) Using the Monte Carlo generator we generate a training list T of 1000 sample points. Recall, a sample point is an instantiation of all of the variables in the graph.

(4) Using the training list T we parameterize each of the graphs G1 through G5 to obtain the distribution P_1 to P_5.

(5) Now we can calculate the relative Bayes error for each of these estimated distributions, $RBE(P_i,P_t,Y)$ for $1 \leq i \leq 5$.

(6) Since the Bayes error will vary depending upon the true distribution we use the Delta Bayes error (DBE) to describe the deviation in classification error for the different distribution relative to the true distribution. (where $DBE(P_i,P_t,Y) = RBE(P_i,P_t,Y) - BE(P_t,Y)$). This sequence of operations (steps (1) through (6)) is carried out 20 times.

Experiments 2-5 - These experiments differ from experiment 1 only in that we assume that GX is the true graph in experiment X.[9]

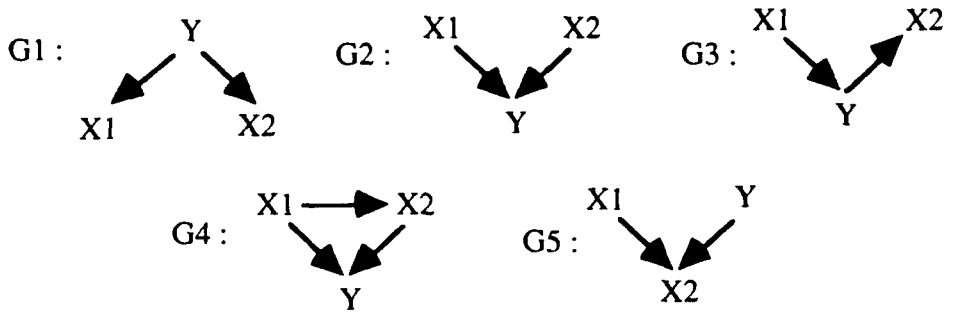


Figure 7

§3.3 Empirical Results

The experimental results given in (figure 8) indicate that structure does matter up to a point. For each experiment we give the mean delta Bayes error and standard deviation (sd) for the delta Bayes error for each graph. The bold numbers in each "mean DBE" row are the lowest values obtained. In each case, the graph with the correct structure did at least as well as any of the other graphs. The average penalty for classifying with the incorrect graph was an increase in expected classification error from 0.0 to 6.9 percentage points. Observe the pair of the columns G1 and G2 for each experiment. The results are identical. This is not surprising since the independence constraints imposed by the topology of these graphs are identical. Now observe the pair of columns G2 and G3. These too are identical. In this case, the independence constraints imposed by the two graphs are not the same. A general theory as to what structure does matter has been worked out but is too complicated to discuss here. (See [Meek 93])

[9] We have run these experiments with different sample sizes and graphs on four variables with similar results.

Experiment 1

	G1	G2	G3	G4	G5
Mean DBE	**0.0007**	0.0010	**0.0007**	0.0010	0.0697
sd of DBE	0.0013	0.0018	0.0013	0.0018	0.0539

Experiment 2

	G1	G2	G3	G4	G5
Mean DBE	0.0318	**0.0025**	0.0318	0.0025	0.0568
sd of DBE	0.0301	0.0025	0.0301	0.0025	0.0565

Experiment 3

	G1	G2	G3	G4	G5
Mean DBE	**0.0016**	**0.0016**	**0.0016**	**0.0016**	0.0691
sd of DBE	0.0043	0.0036	0.0043	0.0036	0.0791

Experiment 4

	G1	G2	G3	G4	G5
Mean DBE	0.0490	**0.0029**	0.0490	**0.0029**	0.0481
sd of DBE	0.0415	0.0038	0.0415	0.0038	0.0426

Experiment 5

	G1	G2	G3	G4	G5
Mean DBE	0.0535	0.0017	0.0535	0.0017	**0.0016**
sd of DBE	0.0593	0.0027	0.0593	0.0027	0.0026

Figure 8

§4 Conclusions and Further Research

In the preceding section we have given empirical evidence for the claim that network structure matters for classification and prediction. It was shown that using the correct structure to parameterize a Bayesian network improves the classification accuracy of Bayesian networks. This is empirical evidence in support of using reliable methods for the discovery of network structure (topology). As mentioned earlier, the network constructor in TETRAD II is asymptotically reliable under weak assumptions. The results of these experiments shows that structure does indeed matter at least for stochastic neural networks. The informal argument given in §1.1 about the importance of network topology for classification with neural networks combined with the empirical evidence suggests that the topology for all types of neural networks is important for prediction, classification and recognition.

This leads to several open research questions. Given the natural connection between neural networks and Bayesian networks, how can one adapt techniques for constructing Bayesian networks to the domain of neural networks? Can neural network construction techniques improve Bayesian network construction methods especially in the case where there are latent variables? Are there more general techniques for the construction of network topology? In addition, much work needs to be done to formalize the analysis of both cyclic neural networks, cyclic Bayesian network and the connections between the two.

There are several issues that need to be addressed about prediction and classification with both Bayesian and neural networks. For instance, what is the variance of the prediction or classification for training sets of different sizes for both Bayesian and neural networks? In addition, which classification, prediction and recognition domains are particularly well-suited for the Bayesian network technology and which are better suited for neural network technology?

§5 References

Ash, T. (1989). "Dynamic node creation in backpropagation networks." ICS Report 8901, San Diego, CA: University of California.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees*, Wadsworth, Belmont.

Freeman, J. and Skapura, D. (1991) *Neural Networks*, Addison Wesley, New York.

Frean, M. (1990). "The upstart algorithm: A method for constructing and training feedforward neural networks." Neural Computation, 2, pp. 198-209.

Herskovits, E. (1991). *Computer-Based Probabilistic-Network Construction*, Ph.D. dissertation, Stanford.

Kamin, E.D. (1990). "A simple procedure for pruning back-propagation trained neural networks." IEEE Trans. on Neural Networks, 1, pp. 239-242.

Meek, C. (1993) Classification and Bayesian Networks, Forthcoming.

Spirtes, P., Glymour, C., and Scheines, R. (1993). *Causation, Prediction, and Search*, Springer-Verlag.

Spirtes, P., Glymour, C., Scheines, R., Meek, C., Fienberg, S., Slate, E. (1992). "Prediction and Experimental Design with Graphical Causal Models." Tech.Rept. CMU-PHIL-32.

Applications of SLEUTH to Navy Manpower, Personnel and Training Data

Janice D. Callahan[1] and Stephen W. Sorensen[2]

Abstract

The Navy Personnel Research and Development Center (NPRDC) developed an automated exploratory technique for categorical data, SLEUTH, that locates artifacts in large data sets. SLEUTH's accuracy rate is over 75% on published problems. We describe the development of SLEUTH, drawing lessons we learned from two existing programs. We apply SLEUTH and those two programs to Navy problems much larger than any in the literature and discuss the new evaluation methods required for large data sets.

Purpose

Examples of Large Data Bases

The United States Navy, like every organization, collects data as part of the normal course of business. For example, the Navy Integrated Training Resources Administration System contains data on each training course the Navy teaches and each student going through the courses. For research purposes, the Navy Personnel Research and Development Center (NPRDC) restructures the data into a more useable form. The student data is collected into a file called TRAINTRACK that is a longitudinal historical data base of training incidents by Social Security Number. The training history of any Navy person from recent enlistee to admiral can be accessed by typing in his or her Social Security Number. Currently TRAINTRACK contains over 1,500,000 records and 65 elements per record. The file can be used to answer research questions such as the causes of attrition from training.

Other data sets are collected for special purposes and can be

---

[1]Callahan Associates Incorporated, San Diego, CA.

[2]Navy Personnel Research and Development Center.

This research was partly performed under contract N66001-91-D-9507 that Systems Engineering Associates, San Diego, CA has with the Navy Personnel Research and Development Center. The opinions expressed in this paper are those of the authors, are not official, and do not necessarily reflect the views of the Navy Department.

used for research. The Youth Attitude Tracking Survey (YATS) is an annual survey of 18 to 25 year olds on their likelihood of enlisting in the armed forces. Over 10,000 individuals are surveyed each year, and each is asked up to 500 questions. To assist in research, the Social Security Numbers are matched with later enlistees. Using the file and the later match, a researcher can investigate questions on advertising effectiveness and other influences on enlistment.

Applying the Scientific Paradigm

How should a researcher approach these large files? The usual scientific paradigm is that a researcher forms a hypothesis about relationships (such as cause and effect). Then the researcher decides the data that he or she needs to test the hypothesis. Next the data is collected. Statistical techniques applied to the data allow the researcher to accept or reject the hypothesis. If possible the researcher builds a model from the data. An accepted (or rejected) hypothesis and a model advance the theory. Finally in an attempt to extend theory, the researcher forms a new hypothesis and begins the cycle again. A significant feature of the scientific paradigm is that the researcher selects his or her own data. Both the data selection and the hypothesis are necessary to advance theory.

One way to follow the usual scientific paradigm on large pre-existing files is the following: Form a hypothesis that can be answered by data in the file. Then continue as before. Extract the data from the file, statistically test the hypothesis, build a model, and advance theory. Unfortunately, following this paradigm on pre-existing files means that the hypotheses and theory must necessarily become skewed. Many important hypotheses will never be tested because a hypothesis must be limited by the data at hand.

For example, using TRAINTRACK the researcher has no influence on what data are collected; those choices are made for the purpose of Navy management. All that the researcher can do is decide which data elements to extract from the management files. For the case of YATS, a researcher involved in setting up the survey can pick questions to ask. But in practice most military researchers get the data after the survey is completed and have no input in the structure of the questionnaire.

On the other hand, the data in large pre-existing files are almost certainly greater than anything a researcher will collect alone. Consequently many additional hypotheses are available for testing if the researcher can imagine them. But practical difficulties with this approach include:

1) Most statistical techniques and model structures involve a limited subset of variables (usually fewer than 10).
2) When the researcher picks variables from the data set,

human biases and preconceptions are sure to intrude.

3) And, the analysis of a large data set is a labor intensive process that quickly tires the most dedicated researcher.

Exploring a New Paradigm

So we come to a new scientific paradigm: Use statistical and computer techniques to automate the process of forming a hypothesis, testing a hypothesis and building a model. Apply the automated techniques to the entire data set. The researcher then analyses the results of the automated technique and uses those results to establish theory. This paradigm is usually dismissed with the pejorative "empiricism", but the practical reality of the existence of these large data bases in the Navy and elsewhere mean that the new paradigm must be examined.

Under the new paradigm the researcher's domain knowledge, including knowledge of theory, is replaced by the computer's ability to do a total or near-total search. The computer formulates and tests a large set of hypotheses or models -- all of those in the class that it was designed to consider.

The goal of this paper is to examine the question: Can a completely automatic discovery system develop good explanatory models in large data bases?

Methods

This 1993 Neural Network Conference contains many papers that use neural networks to build models on non-linear data. For the most part, these models are extremely accurate, for reasons that are only now being understood. The neural network models are not yet very interpretable, although several researchers are progressing in this area. Consequently neural networks seem best suited to applications that emphasise accuracy rather than explanation.

This paper focuses on interpretable models and that takes us into the field of symbolic processing which sometimes rivals and sometimes complements neural networks.

To explore the question about whether a completely automatic system can build good models, we describe the application of three systems to Navy data sets. Two of the systems, CART and TETRAD II, were built outside the Navy (although recent development on TETRAD II was partly funded by the Navy). The third system SLEUTH was built at NPRDC based on our experiences with the other two systems. We apply the three systems to Navy data bases and evaluate them based on the interpretablity of the results, the richness of the models, and their ease of use.

Results

Classification and Regression Trees (CART)

Breiman, et.al. (1984) developed Classification and Regression Trees (CART) to apply tree methods to classification and later regression. Earlier work along the same line included Automatic Interaction Detection (AID) and a revised program, THAID, at the Institute for Social Research, University of Michigan. Breiman, et.al. argued that a data set is interesting not only because of its size, but also because of its complexity. Complexity includes high dimensionality, a mixture of data types, nonstandard data structure, and non-homogeneity (where different relationships between the variables hold in different parts of the space). They believed that trees captured much of this complexity.

The CART work is very similar to the famous ID3 algorithm in machine learning (Quinlan, 1986). The principal difference seems to be that CART has a stronger statistical foundation. Meyrowitz (1991) states that the accuracy of models built with ID3 is comparable to the accuracy of models from neural networks.

We were faced with a problem of analysing a group decision-making process consisting of complex data and a changing group of decision makers. Each year Navy planners meet to determine advanced skill training requirements for over 1200 courses. The planners include budgeters, individuals who assign personnel to training, representatives of the training commands, and managers who set the shape and strength of the personnel force structure.

We wanted to understand the decision process that was occuring in the hope of improving the quality of the decisions and shortening the time required to make a decision. But we also wanted to set the stage for replicating the decisions of the group during the course of the year and assisting other groups in later years.

The decision trees in CART seemed strikingly similar to the trees in expert systems. We believed that if we could capture the decision-making in a tree, then we could incorporate it into an expert system. In Callahan & Sorensen (1991) we described our application of CART.

The planners consider data that describe present and future requirements in the fleet for different personnel skills, the current inventory by paygrade, previous training plans and previous utilization of training courses. They also bring to the conference personal information that is not found in the data, such as whether they are getting phone calls from the fleet about shortages in the skill. Figure 1 shows a regression tree using their input data and using their decisions as a response variable. The tree sets the plan based on existing data that the planners had during one year.

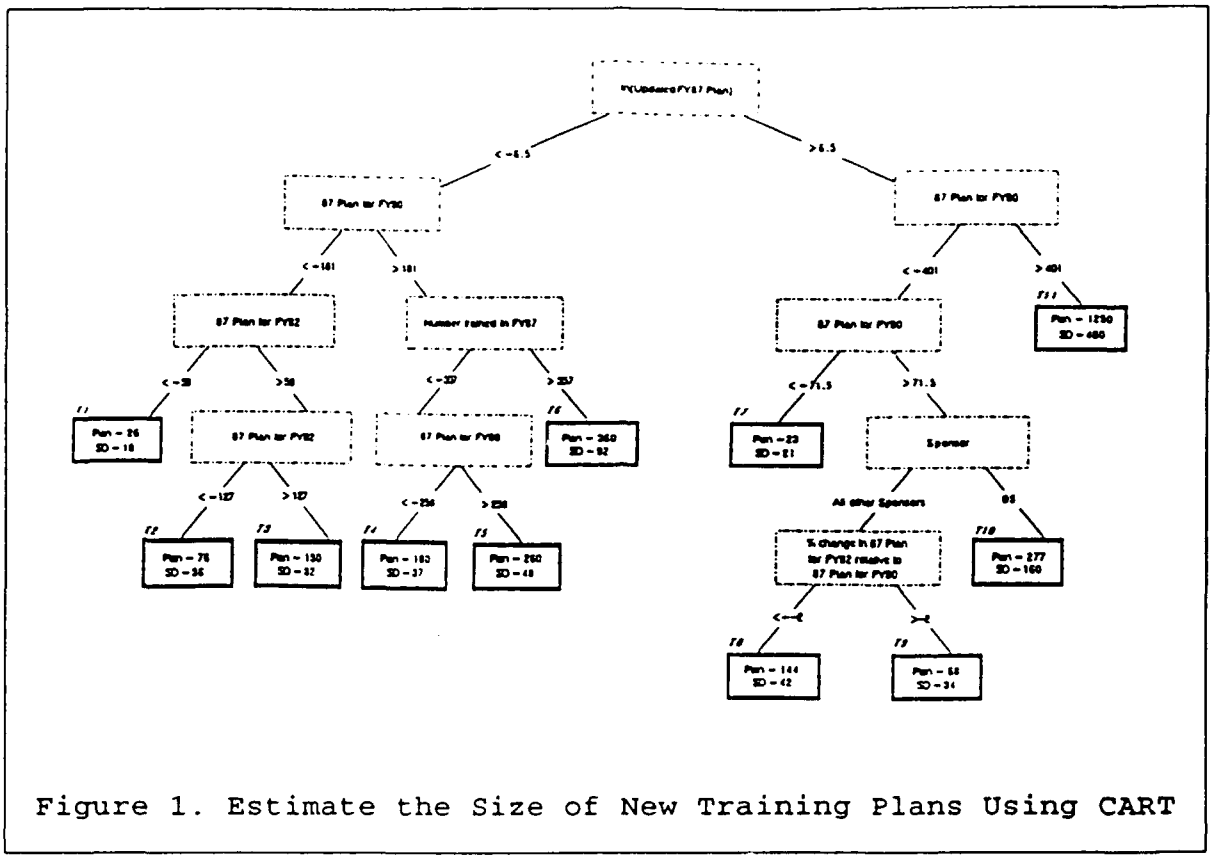


Figure 1. Estimate the Size of New Training Plans Using CART

In the task of analysis we realized that the planners seemed to be anchored on the previous year's plan -- they used it over 40% of the time in setting a new plan. We applied the classification tree methodology in CART to develop a tree (Figure 2) that shows when the planners used the anchor and when they set another figure as the plan.

We learned several lessons from our experience with CART. First we were impressed with its accuracy in a complex and nonlinear situation. It accurately captured the non-homogeneity of the data we were using. Straight regression would not have worked so well. The ability to capture non-homogeneity is easily seen by working down several branches of Figures 1 and 2. Second, we were encouraged by CART's speed. CART builds a full tree and then works backward to get an optimal fit. When we first began thinking how we might build an exploratory system at NPRDC we thought that we would need an alpha-beta algorithm or other techniques to limit search. CART's approach encouraged us to think that we might not have to do that.

The main negative lesson from our experience was that we had great difficulty testing CART on messy data and when some information is not contained in the available data. We never knew whether to blame CART or the data for any problems. We began

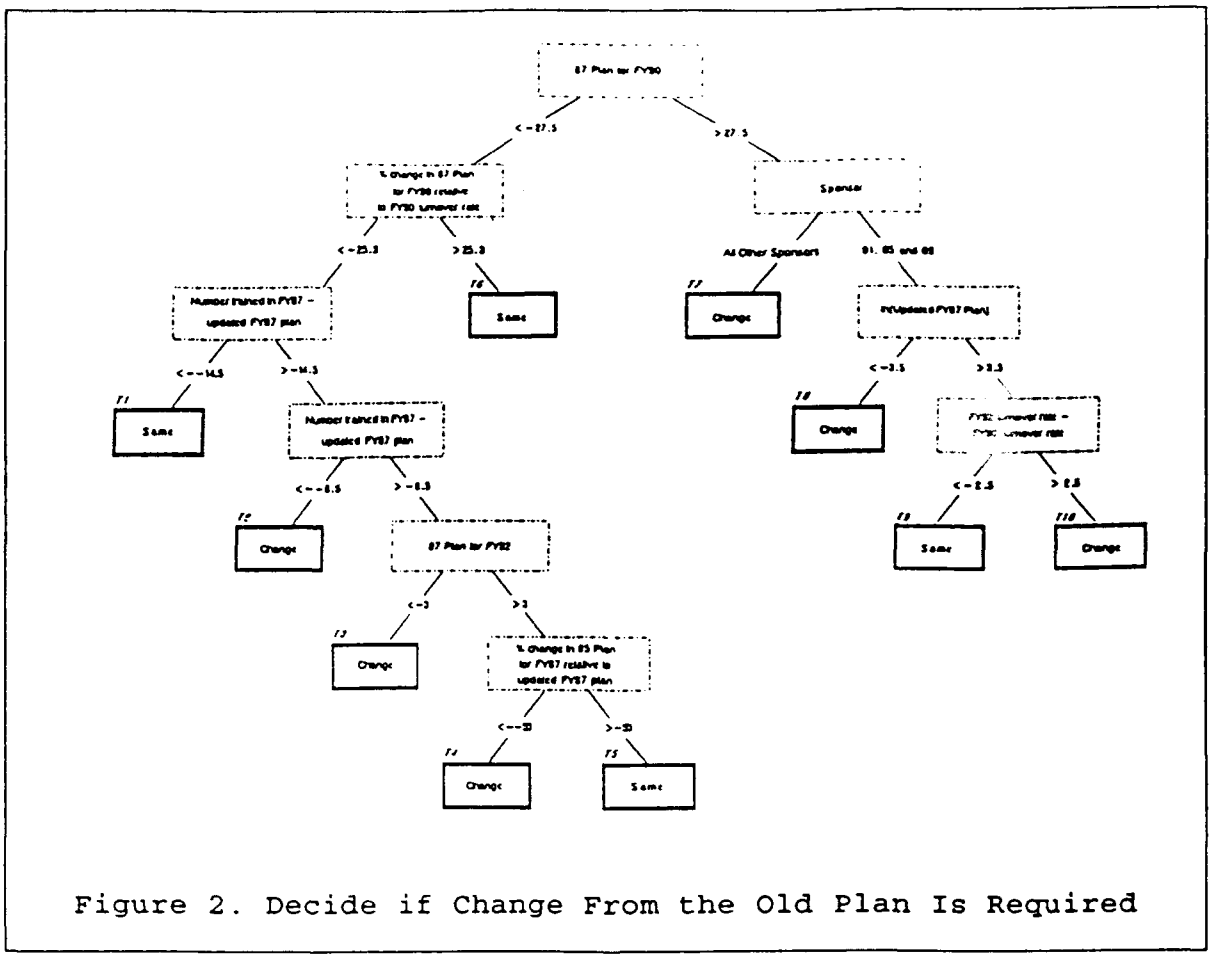


Figure 2. Decide if Change From the Old Plan Is Required

looking for cleaner Navy data to use in the future. A second lesson was to avoid constructed variables (e.g. proportional changes). Constructed variables required exactly the sort of domain knowledge and trial and error effort by the researcher that we wanted to avoid in an automated system. Langley, et.al. (1987) describe programs that build constructed variables, but we believed that we had to postpone that development.

In our later work we have used survey data since it seems cleaner and the rules for constructing variables are more straightforward. This was simple prudence; we needed more confidence in what we were doing before we tackled more complicated problems.

TETRAD II

Next we experimented with the discovery system TETRAD II that was developed at Carnegie Mellon University, in part under a series of contracts with the Office of Naval Research. Spirtes, et.al. (1993) describe TETRAD II as a method to infer causes from statistics. The end result is a set of path models. The program is applicable to both linear and discrete data and can be applied to

a hundred or more variables as long as the causal relationships between variables are sufficiently sparse and the sample is sufficiently large. TETRAD II does not do a total search. It uses an algorithm to decide whether to add new paths to the model and the researcher's domain knowledge to limit the search space.

We were faced with the task of analysing the YATS data set mentioned at the beginning of this paper. Our goal was to determine the influences on an individual's decision to join the military. We used the 1985 questionnaire since that gave sufficient time for respondents to join the military. 854 of the respondents subsequently enlisted and 7625 did not enlist.

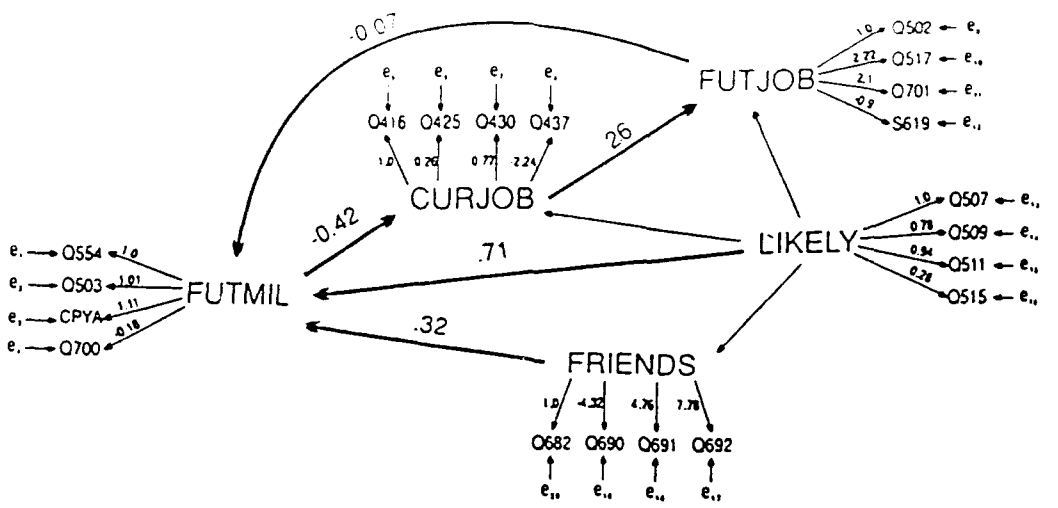
We wanted to run an experiment that completely eliminated a researcher's decision-making from the task of model building. Our strategy was to embed TETRAD II into an automated system that made its own decisions. The experiment is described in detail in Callahan & Sorensen (1992). (Note that more recent developments by the team at Carnegie Mellon University have reduced several of the steps in our process to a single step. Some of this development seems to have resulted from observing our work.)

The first step was to group variables from YATS into seven substantive clusters. These were common-sense clusters of related questions. We later used factor analysis and got slightly worse results. This clustering process was the only step that involved human judgement. TETRAD II analysed the clusters and developed candidate latent variable models. The automated system ranked the candidate models based on output statistics and then gave the highest ranked models to a commercial program (EQS, from BMDP Statistical Software Inc.) for specification. The candidate model with the highest score based on output statistics became the latent variable for that cluster.

The seven latent variables were demographics, current job, likely military, future military plans, future job, friends support the military, and military advertising. Each latent variable contained four indicator variables from the original cluster.

The next step was to build a structural equation model from the latent variables, and the steps were similar to those followed before. TETRAD II produced candidate structural equation models. The highest ranked models based on output statistics were passed to the commercial program for specification. The highest ranked models from the commercial program became our final models.

TETRAD II does not easily handle discrete variables, especially the yes-no type that indicates whether or not an individual taking the YATS questionnaire later joined the military. So we had to split the problem into two groups and look for good models within each group. For those individuals who did not join we got an excellent structural model from our automated system (Figure



FUTure MILitary plans, CURrent JOB, FUTure JOB, LIKELY military, FRIENDS support military

Figure 3. Civilian Structural Model (p=.77) Using TETRAD II

3). The response variable is future military plans and the other variables impact that. The heavy lines in Figure 3 show causal relationships that were consistent in the highest ranked models.

Unfortunately, for the individuals that later joined the military we got models with low scores that were not statistically significant. We speculated that this was because the total population was made up of subpopulations that were influenced by different factors (e.g. friends, current job, advertising).

We were very pleased with the outcome of our experiment. We were able to get good models using a completely automated system. Domain knowledge, regarded as a researcher's greatest asset, was only used to build the common-sense clusters at the beginning and even that step could have been automated by parsing the questions. The selection of the best models at each step was reduced to ranking them by output statistics.

We were also happy about using survey data. Survey data constitutes the type of clean, relatively self-contained universe of information that we needed for future work. The YATS survey had one additional data item added (whether the individual joined the military or not).

We were impressed by TETRAD II and continue to use it in the Navy and encourage its use by others. Nevertheless, we decided that much of the Navy's data in survey and personnel files is categorical data, and TETRAD II cannot easily handle that. We decided to build our own program to explore and model categorical data bases.

SLEUTH

Callahan & Sorenson (in review) describe the first version of SLEUTH. The program is designed to explore for interactions on large multi-dimensional categorical data sets. The goal is to find relationships between variables in the data set; no response variable is required. In the first version of the program, SLEUTH located two-way and three-way interactions. It also isolated variables with no interactions.

SLEUTH processes the input data to discretize the continuous variables and put all variables and data into a contingency table. Then SLEUTH looks at all pair-wise contingency tables conditioned on subsets of data. This generates a large number of Chi-square test results. SLEUTH applies a Bonferroni correction to account for multiple testing. The output orders the interactions using odds ratios for two-way interactions and ratios of odds ratios for three-way interactions.

SLEUTH satisfies the criteria for an exploratory system that we established in early experiments: SLEUTH requires no domain knowledge of the inputs. It works on data common in the Navy's manpower, personnel and training community. SLEUTH does a total search and is very fast. SLEUTH ranks the outputs based on standard statistics. The researcher can apply his domain knowledge to evaluate SLEUTH's outputs.

We validated SLEUTH by testing for false positives and for its abilitiy to match published problems. On 10,000 simulated uniformly distributed data sets, SLEUTH found interactions in only 287 data sets. This conservative result probably comes from the Bonferroni correction. We looked at 21 published examples from Bishop, et.al. (1975) and Agresti (1990). For 11 examples, SLEUTH produced the same results. In 3 other examples, SLEUTH was different but as good. In 4 examples, the published models were more complicated than SLEUTH was designed to locate. SLEUTH got a worse model once, and had two failures.

For a practical test of SLEUTH we used the Navy-Wide Personnel Survey (NPS). This is given annually to Navy enlisted personnel of all ranks and Navy officers below the rank of admiral. The sections of the survey cover personal and career information, issues regarding rotation moves and assignment, recruiting duty, pay and benefits, education and leadership programs, quality of life programs, organizational climate, and AIDS education. Summary

statistics (marginals) are presented to the Chief of Naval Personnel and are reported in Navy newspapers.

In 1991 the survey was distributed to 23,821 individuals and 13,232 completed surveys were returned. The survey contained 95 questions. We studied 31 of them. Our 31-dimensional contingency table contained 7654 non-empty cells. SLEUTH took 5 hours on this problem using a 486/50MHz computer. It found 160 two-way and 165 three-way interactions.

Paygrade was a variable in 62 of the three-way interactions. This seemed reasonable since officers and enlisteds likely have different perspectives on issues, and even within each group careerists may differ from non-careerists. We took this result as evidence that SLEUTH was finding the obvious (an important thing to do). Then for further analysis we broke the data set into two parts: officer and enlisted. A researcher using domain knowledge would have made this break before any other processing. Since we did not assume domain knowledge, we let SLEUTH tell us what to consider.

Table 1 shows the best three-way interaction that SLEUTH found. It is an important example because it shows that an automated exploratory program can point the way to new theory that may be important for Navy managers and for the society at large. The three variables are race, opinion of the detailer's knowledge of available jobs, and the quality of formal leadership training. The detailer is the individual who assigns a sailor to his or her next job. Leadership training has several purposes but one important component is enhancing a sailor's ability to deal with others in the Navy organization.

SLEUTH may have picked out the three-way interaction because of the small numbers in the lower right corners of the contingency tables under Blacks and Other races. However, even if those were not so small, SLEUTH may have picked the interaction because of the very large number (relative to other numbers in the table) in the upper left corner of the table under Blacks. It appears that Blacks who responded favorably to the leadership training also had positive interactions with their detailers. This indicates that leadership training plays a much more important role for Blacks than for Whites in integrating them into the Navy's culture. The issue should probably be investigated further.

We are very encouraged by SLEUTH's results and we continue to develop it along several lines. We found the reason why SLEUTH sometimes got the wrong model on published problems and we corrected it. SLEUTH now finds four-way interactions. We continue to improve SLEUTH's processing speed and are changing it to work on much larger data sets.

Table 1

Best Three-Way Interaction From NPS

Whites		
Opinion of your detailer's knowledge of available billets	How would you rate the quality of the formal leadership training you received in the last class you attended?	
	Does not apply, have not had any, fair, good or very good	Poor or very poor
No opinion, neutral, positive or very positive	4037	136
Negative or very negative	549	38
Blacks		
No opinion, neutral, positive or very positive	1002	11
Negative or very negative	102	6
Other races		
No opinion, neutral, positive or very positive	683	10
Negative or very negative	93	0

Conclusions

We set out to explore the question of building a completely automated exploratory system to construct models on large data sets. Such a system goes against the usual scientific paradigm and requires patience and tolerance on the part of researchers to even entertain the notion. But the reality of large, pre-existing data sets means that the possibility of using exploratory systems must be considered.

In this paper we showed that completely automated exploratory systems exist now and are already very powerful. Not only do they

match published results but also they tackle problems much larger than anyone has considered before. Furthermore they work in reasonable amounts of computer time. They do not require domain knowledge.

We also showed some of the limits of automated exploratory systems. They work best on clean data sets. They can only tackle a problem where the entire domain of the problem is encompassed within the data. Some technical issues, such as constructed variables, are several years away.

References

Agresti, A. (1990). Categorical Data Analysis. New York: John Wiley & Sons.

Bishop, Y. M. M., Fienberg, S. E., & Holland, P. W. (1975). Discrete Multivariate Analysis. Cambridge, MA: The MIT Press.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and Regression Trees. Belmont, CA: Wadsworth International Group.

Callahan, J. D., & Sorensen, S. W. (1991). Rule induction for group decisions with statistical data -- an example. Journal of the Operational Research Society, 42(3), 227-234.

Callahan, J. D., & Sorensen, S. W. (1992). Using TETRAD II as an automated exploratory tool. Social Science Computer Review, 10(3), 329-336.

Callahan, J. D., & Sorensen, S. W. (in review). An exploratory categorical data analysis technique. Applied Statistics.

Langley, P., Simon, H. A., Bradshaw, G. L., & Zytkow, J. M. (1987). Scientific Discovery: Computational Explorations of the Creative Process. Cambridge, MA: The MIT Press.

Meyrowitz, A. L. (1991). Neural networks: a computer science perspective. Naval Research Reviews, 43(2), 13-18.

Quinlan, J. R. (1986). Induction of decision trees. Machine Learning, 1, 81-106.

Spirtes, P., Glymour, C., & Scheines, R. (1993). Causation, Prediction, and Search. New York: Springer-Verlag.

Neural Network Models of Decision-Making Schemas

David E. Smith
Decision Support and Artificial Intelligence Branch
NCCOSC (NRaD)

Sandra P. Marshall
Department of Psychology
San Diego State University

Abstract Research into human cognitive processes involved in tactical decision-making indicates a "naturalistic" model, in which situation assessment receives greater emphasis than course of action selection. Schema theory provides a logical framework for analyzing situation assessment processes. This paper describes the basis for a feed-forward/feed-lateral Knowledge-Based Artificial Neural Net (KBANN) model of decision-making schemas.

Introduction

This paper presents preliminary results of ongoing research into neural net models of decision-making schemas. Its primary objective is to apply new theoretical findings from cognitive science research to derivation of novel principles for design of future decision aids. Previous research has resulted in a new theory of schema development and implementation (Marshall, 1991a; Marshall, 1991b). The current research is an effort to advance schema theory, such that it can be applied within the context of naturalistic decision model (Zsambok, & Klein, 1992) using a Knowledge-Based Artificial Neural Network (KBANN) approach (Towell, & Shavlik, 1990; Towell, & Shavlik, 1992). The result will be manifested in the form of decision aid principles developed for the Tactical Decision Making Under Stress (TADMUS) program.

Naturalistic Decision Theory

TADMUS was initiated following the incidents involving the USS Stark and the USS Vincennes. The objectives of the TADMUS program are to improve our understanding of how decisions are made in combat and to apply recent developments in decision theory, individual and team training, and information display toward enhancing the quality and timeliness of tactical decision making. Motivation for TADMUS arose partly out of concern over a separation between research conducted on decision making and development of tactical decision aids. This resulted in an explicit effort to represent advances in cognitive science in TADMUS decision aid principles.

A naturalistic decision-making model is central to formulation of TADMUS decision aid principles. An early part of the TADMUS program involved a set of reports considering the scope of naturalistic decision making as it occurs in the situations addressed by TADMUS (Kaempf, Wolf, Thordsen, & Klein, 1992; Klein, 1990; Zsambok, et al., 1992) . The nucleus of Klein's research is predicated on a belief that cognitive functions elicited in natural settings involves processes that are likely to differ from those found in artificial and

contrived situations, such as psychological laboratories. A key theme, therefore, is "that results from sterile and contrived situations may not generalize to less constrained and more natural environments (Salthouse, 1992)."

Four striking findings emerge from the Klein analyses. First, usual decision theory does not apply in this context. Their study of experts' protocols suggests that only rarely can one find strategies such as those reported in the psychological decision-making literature. Second, just two types of decisions are made: those involving situation assessment and those involving selecting a course of action. Third, the Klein data and analyses overwhelmingly indicate that only situation assessment presents difficulties to decision makers under stress. That is to say, course of action decisions do occur, but are relatively routine if the situation has already been successfully diagnosed or assessed. Fourth, within situation assessment, two cognitive strategies -- feature matching and story generation -- predominate over all others.

Feature matching is a situation assessment and diagnostic strategy in which features of the current instance lead to situation recognition based on retrieval of prior cases having the same features. This retrieval is then used to adopt an hypothesis or to select between hypotheses concerning the nature of the situation. Not all of the steps involved in this strategy are performed deliberately and consciously; the distinction between perceptual and cognitive processes is not overt. In addition, this strategy ignores a causal context for the evidence. Instead, feature matching is strongly reliant on spatio-temporal relationships between observed events.

A second strategy involved in situation assessment and diagnosis is story generation, which can be described as construction of a causal model for the purpose of inferring how a current situation might have arisen out of an earlier state. The feature matching strategy described above relies heavily on an ability to match a set of features extracted from the environment with a set of features retrieved from memory (in the form of prior cases). This implies that the extracted features are assembled into a pre-existing structure. Story generation is used in cases where such a pre-existing structure is not (readily) available. There may be uncertainty or ambiguity related to the situation, or the situation may be judged as unfamiliar, either condition resulting in an inability to assemble extracted features into the form of a pre-existing structure.

Schema Theory

One of us has recently completed a long-term project about the nature of problem-solving schemas (Marshall, 1991a; Marshall, 1991b; Marshall, in press a; Marshall, in press b). The result of that ONR-sponsored research is a new theory of schema development and implementation. The theory stipulates that four components of schema knowledge may be identified and assessed. These are identification knowledge, elaboration knowledge, planning knowledge, and action knowledge. Each is described briefly below.

The central function of *identification knowledge* is pattern recognition. It is this knowledge which contributes to the initial recognition of a situation. Pattern recognition occurs as a result of the simultaneous cognitive processing of many features: no single feature serves to trigger the recognition of a situation. Rather, different configurations of several features present different patterns, and they must all be recognized as the same basic situation, depending on the specific characteristics that are noticed.

Elaboration knowledge enables the construction of an appropriate mental model of the situation. Here we find not only the general structure of the situation but also descriptions of its components, i.e., the details about what is allowable, what is not, and how all of the necessary pieces accommodate each other. Almost certainly the individual needs to have in memory an example situation to serve as a baseline analogy against which he or she can match the current problem components, in order to evaluate the fit of the hypothesized situation as determined through the constraint knowledge. The general form of the mental model may come from a specific example or may derive from a generalized description of a situation.

Planning knowledge contains information about how to identify any unknown part(s) of the situation and is instrumental in formulating immediate goals and subgoals. Planning knowledge is frequently very difficult knowledge for individuals to acquire. It depends greatly on having the appropriate mental model of the current situation and using that model comfortably.

Planning knowledge is used to determine which steps to take in solving a problem. *Action knowledge* follows up on the plan by carrying out those steps. As each piece of the plan is completed, the execution knowledge is called on to address subsequent ones.

Knowledge-Based Artificial Neural Nets

Previous research on artificial neural networks (ANNs) has until recently paid little attention to existing domain knowledge in determining ANN topology. Application of domain knowledge has been limited to design and development of input and output vectors and to construction of training, test, and validation sets. Shavlik, et al., (Maclin, & Shavlik, 1991; Towell, Craven, & Shavlik, 1991; Towell, & Shavlik, 1990; Towell, & Shavlik, 1992) have developed a methodology for using domain knowledge to determine both network topology and initial weight values. The result is a knowledge-based artificial neural network (KBANN) which explicitly represents domain theory and starts with weights significantly better than random.

The KBANN algorithm applies a knowledge base of domain-specific inference rules to design a network topology and initial weights. Towell supplies the following example in (Towell, et al., 1991).

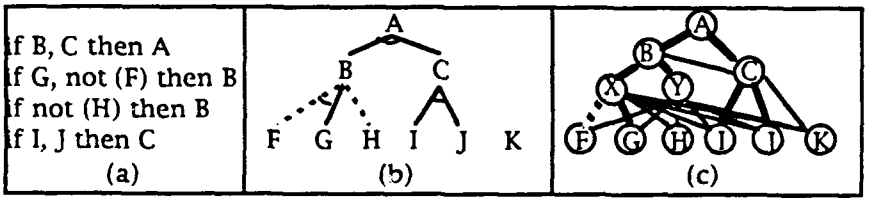


Figure 1. Translation of a Domain Theory into a Knowledge-Based Neural Network (KNN)

Figure 1(a) presents an artificial domain theory defining membership in category A. Figure 1(b) is a hierarchical representation of these rules: solid lines and dotted lines representing necessary and prohibitory dependencies, respectively. Figure 1(c) represents the resulting KNN. Units X and Y in

figure 1(c) are used in the KBANN algorithm to handle disjunction in the rule set. Other than these nodes, each unit corresponds to a consequent or an antecedent in the domain theory. Thick lines represent heavily-weighted links, corresponding to necessary dependencies expressed in the domain theory. Dotted lines are prohibitory links, and thin lines are links added to allow refinement of the domain theory.

It is not necessary for the domain theory to be either fully complete or correct. It is only required that the domain theory support approximately correct reasoning. The domain knowledge, once translated into the KNN, will be modified and made more robust through training of the network. This is represented in figure 2. After training the network, an improved form of the domain theory can be formulated from rules extracted from the network.

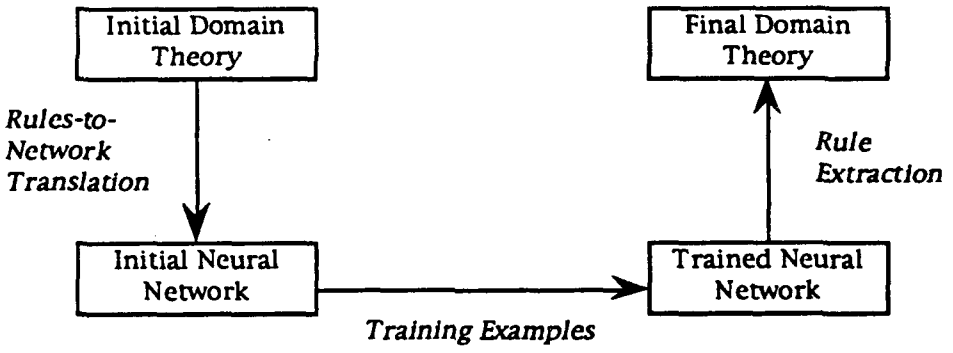


Figure 2. KBANN Information Flow

This methodology makes tow assumptions concerning the nature of the network(Towell, et al., 1991). The first is that the meaning of the units (nodes) is not significantly shifted or altered by training the net. Thus, domain labels applied in creation of the network continue to correspond to extracted rules. The second assumption is that activation values in the trained net are near either one or zero. The rationale for this assumption is that it allows each non-input unit to be treated as either a step function or a Boolean rule.

Both of these assumptions are valid within the context of the KBANN models constructed by Shavlik, et al. (Maclin, et al., 1991; Towell, et al., 1991; Towell, et al., 1990; Towell, et al., 1992) The first assumption, regarding unit meaning, will remain valid here. However, the second assumption is no longer valid in this setting. The AAW domain requires not only real-valued inputs (e.g., range, speed, altitude), but real values for the units internal to the net. These nodes represent pieces of information which take on real values in the domain being modeled.

Decision-Making Context

Inasmuch as the original events motivating the TADMUS program involved antiair warfare (AAW), the program was constructed to obtain a thorough understanding of the tasks required in AAW. Toward that end, several scenarios were assembled which place a six-member Aegis ship combat information center (CIC) crew in simulated combat situations. The six members involved are the Commanding Officer (CO), Tactical Action Officer

(TAO), Antiair Warfare Coordinator (AAWC), Tactical Information Coordinator (TIC), Identification Supervisor (IDS), and Electronic Warfare Supervisor (EWS). The scenarios are presented on a time-step scenario generation facility known as the Decision-making Evaluation Facility for Tactical Teams (DEFTT). DEFTT comprises six personal computers on an ethernet local area net driven by a Hewlett-Packard 9000-series host. System performance at watchstation consoles (the six personal computers) is similar to that found in the CIC on an Aegis ship.

For additional information on this facility, see (Hutchins, & Duffy, 1992).

KBANN Model of Decision-Making Schemas

The incorporation of schemas into TADMUS offers a number of advantages. First, schemas provide a logical framework for analyzing situation recognition by decision makers. Klein and his associates have already provided important information concerning the nature of situation assessment. They found that situation assessment was the result of either feature matching or story generation. Both of these strategies are easily derived and explained under schema theory, in that we theorize that each originates from a different aspect of schema knowledge. On the one hand, feature matching is a natural outcome of the application of identification knowledge. The co-occurrence of identifiable features occasions recognition of a situation. On the other hand, story generation depends primarily on elaboration knowledge and its associated mental model. When incoming features are insufficient to produce recognition using identification knowledge alone, a mental model may be called upon to provide the underpinnings of a story that is consistent with the observed features and that can supply default characteristics for any missing data. This mental model is the one associated with the situation best approximated by the features. The "best guess" from the identification knowledge allows access to the mental model. Thus, feature matching allows all possible features to have influence, while story generation looks for particular features that match the story reflected by the mental model of the schema and allows inferences about their origins and/or consequences.

An additional advantage of the schema approach is that it allows explicit computer modeling of each cognitive strategy. The importance of the modeling is that it allows us to observe the essential components of the decision and how they are related. This, in turn, allows the formation and analysis of hypotheses concerning the absence of one or more of the components.

A final advantage is that this method allows us to model the schemas of many different experts and to synthesize their approaches.

The first step in creation of KBANN models of cognitive schemas is to generate knowledge networks and cognitive maps representing knowledge base rules elicited from domain experts. Each knowledge network consists of a set of nodes, representing distinct pieces of information provided by the experts, and links connecting the nodes, representing associations between the pieces of information. A larger cognitive map is produced by connecting knowledge networks at common nodes an links, as indicated by relationships specified domain rules elicited from domain experts.

An example of this procedure is provided in figure 3, showing that there exist relationships between altitude, range, bearing, speed, course, and location.

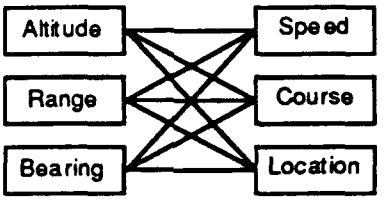


Figure 3. Example Knowledge Network 1

A second knowledge network is represented in figure 4, representing additional domain rules.

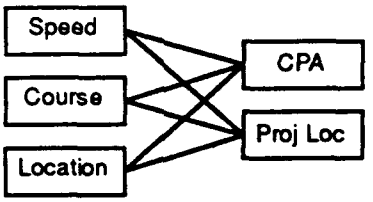


Figure 4. Example Knowledge Network 2

These two example knowledge networks may obviously be linked through other existing domain rules to produce a larger knowledge network, representing a cognitive map showing the information paths connecting, or associations relating, different pieces of information. This is depicted in figure 5.

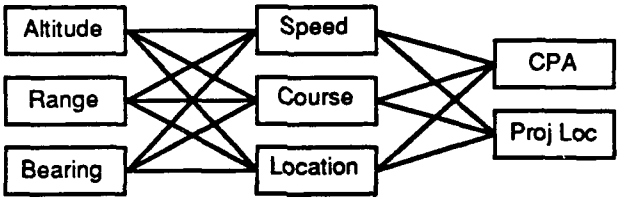


Figure 5. Example Cognitive Map

At this point, we should note that these somewhat overly simplistic and obvious examples are not intended to mislead. They are merely intended to illustrate the concepts involved in translating domain theories into a KBANN. In the first place, the examples provided lend themselves well to solution by known algorithmic methods. We do not propose bypassing or rejecting such solutions. At the same time, we desire to have the inputs to the net represent data available to a decision maker as closely as possible. However, this does not mean that we intend to train a neural net to perform algorithmic tasks. Instead, at algorithmic nodes we will use activation functions duplicating the functionality of the algorithm, rather than applying a more traditional sigmoid (or other) activation function.

Second, the examples provided so far do not demonstrate the complexity of the domain (AAW). Some idea of the complexity involved is given in figure 6.

This example illustrates the three components of the complexity of the domain: multiple information dependencies, Hopfield-like operation in one portion of the net, and recurrence.

The input layer (the leftmost layer of circular nodes) receives data from the environment. The first hidden layer (the second layer of circular nodes from the left) represents information derived or computed from the input layer. The second hidden layer represents further information computed on the basis of data input to the net, but *not* computed directly from input data. The next layer to the right represents knowledge obtained from the pattern of activity at lower levels. The pattern of activity output from this layer indicates the assessment, or diagnosis, of the situation. The single node remaining, to the right of the situation assessment layer, represents potential courses of action to be taken on the basis of the situation assessment.

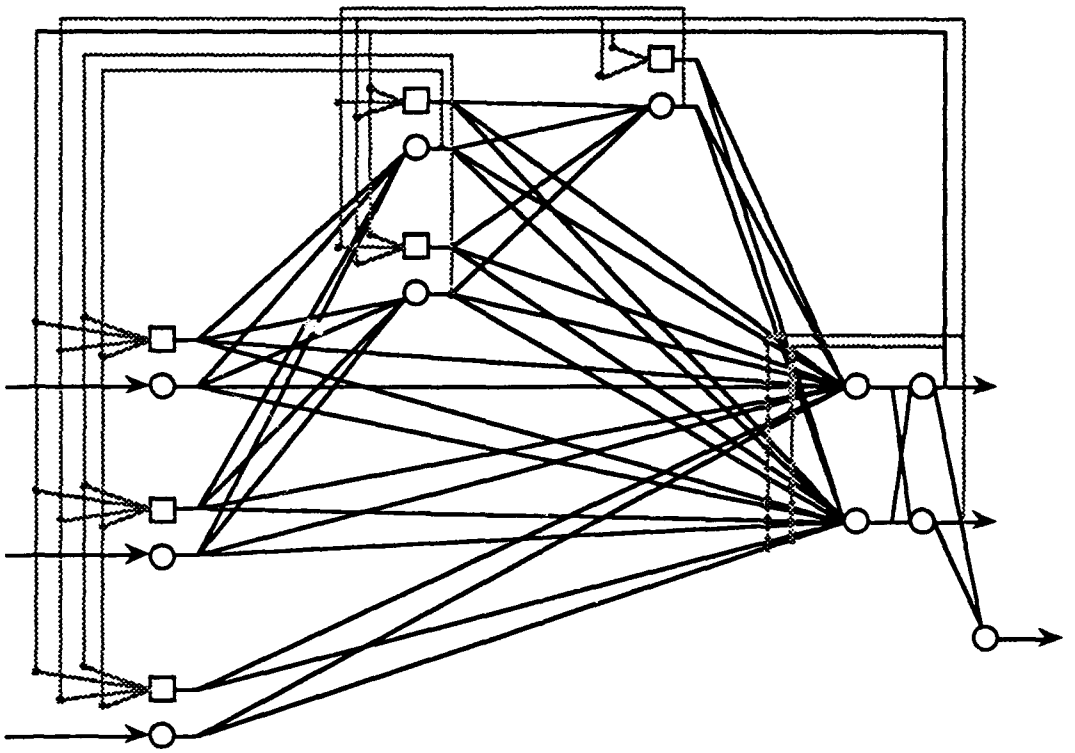


Figure 6. Example Domain Complexity

The second component to be considered is the operation of the situation assessment layer of nodes. These nodes are depicted in figure 7 (multiple inputs are represented by single lines for simplicity). In some respects, operation of this layer resembles a Hopfield net, in that the nodes in figure 7 can be represented as shown in figure 8. Nodes 1 and 2 in figure 7 are identical, with activation between them fixed at 1. The same is true for nodes 3 and 4.

The purpose of this topology is to model the psychological behavior of activation spreading within a single conceptual level. In figure 7, nodes 1 and 2 are directly activated by nodes at lower levels in the net. Nodes 3 and 4 are

activated not only by themselves, but also by other nodes at the same level with which they are linked or associated. While this may be similar to a Hopfield layer, the operation of the net is functionally distinct, in that there is a single pass through this feed-forward/feed-lateral layer, the weight matrix is not specified in advance, and the weight matrix is not (necessarily) symmetric about a 0 diagonal.

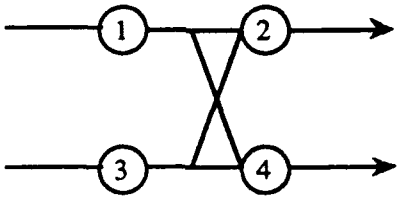


Figure 7. Situation Assessment Layer

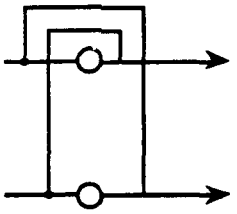


Figure 8. Hopfield-like Representation of the Situation Assessment Layer

The nature of this layer can be made clear by being concrete with respect to the concepts embodied by the nodes in this layer. Figure 9 depicts candidate information nodes in a feed-forward/feed-lateral arrangement. For instance, it can be clearly seen from this illustration that the capabilities of a particular contact are associated not only with lower-level information and data elements, but also with elements within the same level. The pattern of activity within the net thus represents a pattern of cognitive activity.

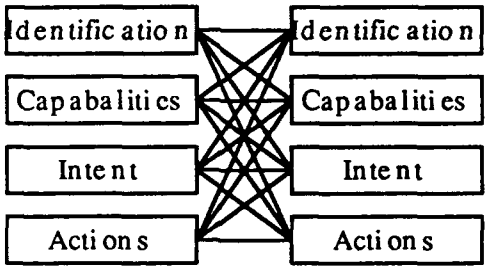


Figure 9. Candidate Feed-Forward/Feed-Lateral Components

The third component of complexity is a requirement to represent time-dependent relationships among pieces of information. This will be

accomplished by creating a recurrent net, thus providing a temporal context for input data from the environment. The recurrent portions of the net are depicted as square nodes and gray association lines in figure 6.

A fully connected candidate version of the net is depicted in figure 10. In this model, an assessment of the situation is represented by a pattern of activity at the output of the feed-forward/feed-lateral layer of the net. The behavior of the trained net will be said to resemble the feature matching cognitive strategy when a situation assessment is produced on the basis of inputs from the environment and existing associations between pieces of information (weights between nodes). The trained net can be said to be performing story generation when there is no dominant pattern of activity at the output of the feed-forward/feed-lateral layer, thus necessitating some amount of adjustment of the associations between pieces of information (weights). An incorrect pattern of activity at the output of the feed-forward/feed-lateral layer does not mean that story generation has been performed. Such an occasion is identical to misidentification of a situation (on the basis of pattern recognition or feature matching) on the part of a human operator. It does mean that the weights should be adjusted, which represents story generation to explain the situation after the fact. Further, repeated failure of a dominant pattern to appear at the output of the feed-forward/feed-lateral layer indicates that the particular situation must be a difficult one for the net to arrive at a "logical" explanation for the evidence.

These types of behavior can be expected to occur as a result of training the net on the basis of a small number of domain experts (small training set). This resembles the brittleness found in expert systems, and severely detracts from a major advantage of neural nets: their ability to generalize. On the other side of this same coin is the fact that a net such as this will be constrained by its size in its ability to store large numbers of patterns. These issues will be resolved by training the KBANN model as discussed above (in the section titled Knowledge-Based Artificial Neural Nets), by using multiple domain experts across multiple tactical scenarios to enlarge the training set, and by constraining the number of patterns to be stored by using only similar scenarios within the AAW domain.

Specifically, if we assume ε: $0 < \varepsilon < 1/8$, Baum and Hassler (Baum, & Haussler, 1988) have shown that if

$$m \geq O\left(\frac{W}{\varepsilon} log \frac{N}{\varepsilon}\right)$$

random examples can be loaded on a feedforward network of linear threshold functions with N nodes and W weights, so that at least a fraction

$$1 - \frac{\varepsilon}{2}$$

of the examples are classified correctly, then the network will correctly classify a fraction $1 - \varepsilon$ of future test examples drawn from the same distribution with confidence approaching certainty.

In the net depicted in figure 10, with 23 nodes and 129 weights, and assuming a maximum error of 1/10, the number of training examples required to achieve this performance is on the order of 3046.

However, the Desired Antecedent Identification (DAID) algorithm, described in (Towell, & Shavlik, 1990), has been shown to decrease the effort required to train a KBANN model to approximately 61% in some domains. While this admittedly may be a best-possible case, application of the algorithm may significantly reduce training set size requirements.

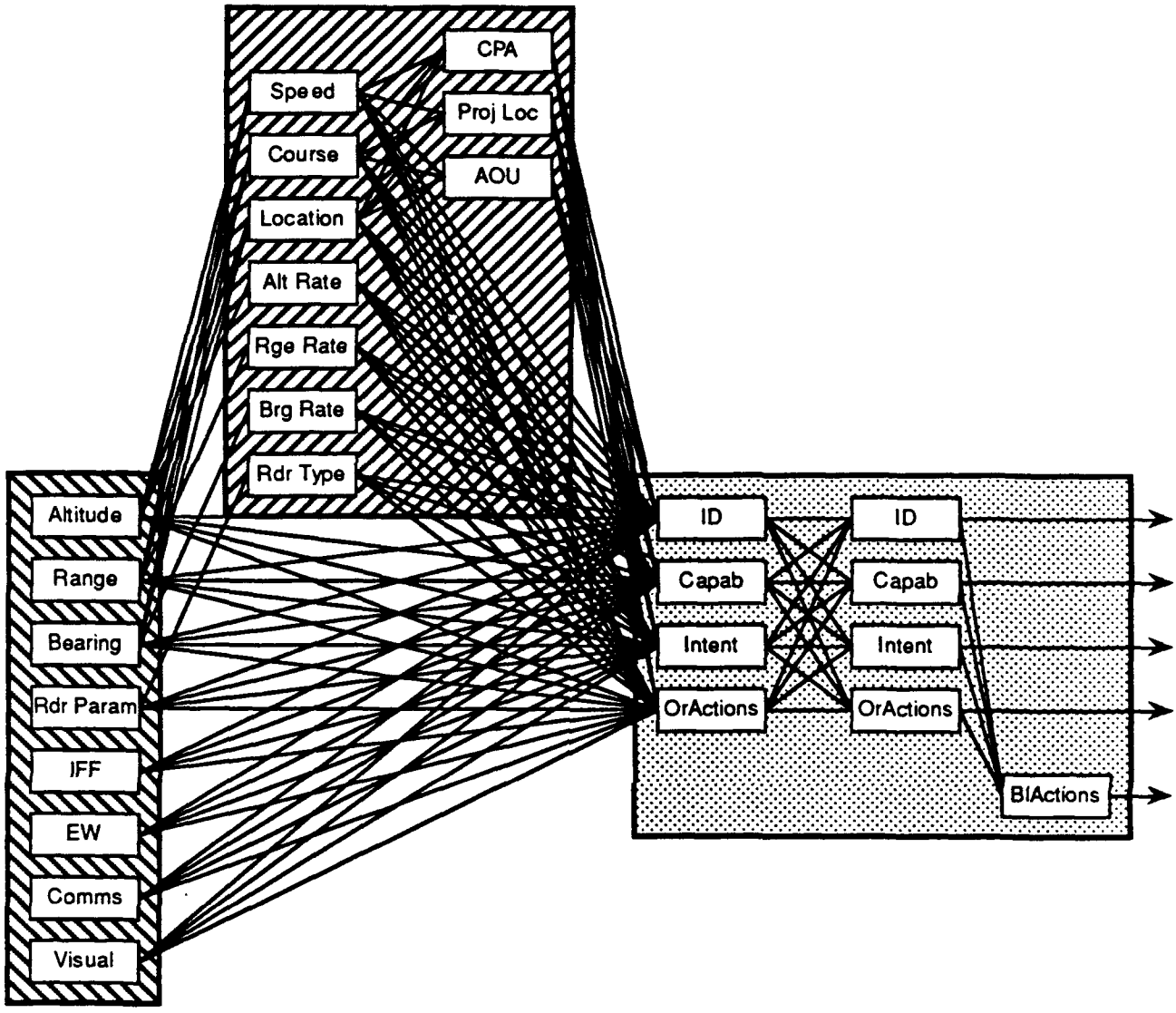


Figure 10. Fully-Connected Candidate Knowledge-Based Neural Network

Summary and Conclusions

An important product of the schema research in problem solving was the development of a methodology by which schemas may be identified, their applications by individuals evaluated, and their component interactions modeled. A modified version of that methodology, involving the generation, implementation, and evaluation of a KBANN model is being used to incorporate decision-making schemas in TADMUS. This will be evaluated for its contributions to a decision support system intended to improve tactical decision making by CIC officers under conditions of stress.

Construction of a neural network model of schemas offers opportunities to gain insight into the nature of cognitive strategies involved in tactical decision making. Further, the same model provides a framework for unifying the cognitive strategies of feature matching and story generation into a single

cognitive context. As such, the model also presents the possibility to advance shema theory from the problem solving domain to decision making.

In addition, construction of schema models representing the decision processes involved in tactical decision making will allow identification of important features of the environment and determination of their significance. This is a direct result of the KBANN methodology, and will be important not only to understanding identification and elaboration knowledge, but also to construction of a feature library, or set of feature objects, for the TADMUS decision support system. These feature objects form the library from which decision makers will be able to construct templates which may be used for particular situations.

References

Baum, E. B., & Haussler, D. (1988). What Size Net Gives Valid Generalization? ,

Hutchins, S. G., & Duffy, D. L. T. (1992). Decision-Making Evaluation Facility for Tactical Teams. Monterey, CA: Ninth Annual Conference on Command and Control Decision Aids

Kaempf, G. L., Wolf, S., Thordsen, M. L., & Klein, G. A. (1992). Decisionmaking in the AEGIS Combat Information Center (Technical Report: Task 1). Prepared for Naval Command, Control, and Ocean Surveillance Center. Fairborn, OH: Klein Associates.

Klein, G. A. (1990). Naturalistic Decision Making. Prepared for Naval Command, Control, and Ocean Surveillance Center. Fairborn, OH: Klein Associates.

Maclin, R., & Shavlik, J. W. (1991). Refining Domain Theories Expressed as Finite-State Automata. Evanston, IL: Morgan Kaufmann, 524-528.

Marshall, S. P. (1991a). Computer-Based Assessment of Schema Knowledge in a Flexible Problem-Solving Environment . Technical Report 91-01, Office of Naval Research Contract No. N00014-90-J-1143. San Diego: San Diego State University.

Marshall, S. P. (1991b). Schemas in Problem Solving: An Integrated Model of Learning, Memory, and Instruction . Technical Report 91-02, Office of Naval Research Contract No. N00014-90-J-1143. San Diego: San Diego State University.

Marshall, S. P. (in press a). Assessing Schema Knowledge. In I. Bejar, N. Frederiksen, & R. Mislevy (Eds.), Test Theory for a New Generation of Tests . Hillsdale, NJ: Erlbaum.

Marshall, S. P. (in press b). Statistical and Cognitive Models of Learning Through Instruction. In A. Meyrowitz, & S. Chipman (Ed.), Cognitive Models of Complex Learning . Norwell, MA: Kluwer Academic Publishers.

Salthouse, T. A. (1992). Cognition and Context. Science, 257, 982-983.

Towell, G. G., & Shavlik, J. W. (1990). Refinement of Approximately Correct Domain Theories by Knowledge-Based Neural Networks. In Proceedings of the Eighth National Conference on Artificial Intelligence. Boston, MA: AAAI Press/The MIT Press, 861-866.

Towell, G. G., & Shavlik, J. W. (1990). Refinement of Approximately Correct Domain Theories by Knowledge-Based Neural Networks. San Jose, CA: AAAI Press/The MIT Press, 861-866.

Towell, G. G., & Shavlik, J. W. (1992). Using Symbolic Learning to Improve Knowledge-Based Neural Networks. In Proceedings of the Tenth National Conference on Artificial Intelligence. San Jose, CA: AAAI P ess/The MIT Press, 177-182.

Towell, G. G., & Shavlik, J. W. (1992). Using Symbolic Learning to Improve Knowledge-Based Neural Networks. San Jose, CA: AAAI Press/The MIT ress, 177-182.

Towell, G. G., Craven, M. W., & Shavlik, J. W. (1991). Constructive Induction in Knowledge-Based Neural Networks. Evanston, IL: Morgan Kaufmann, 213-217.

Zsambok, C. E., & Klein, G. A. (1992). Decisionmaking in Complex Military Environments . (Technical Report: Task 4). Prepared for Naval Command, Control, and Ocean Surveillance Center. Fairborn, OH: Klein Associates.

A Neural-Network Based Behavioral Theory of Tank Commanders

TUNG BUI
The US Naval Postgraduate School
Department of Administrative Sciences
Monterey, CA 93943

Abstract

Based on the presumption that certain data observed in high-tech and fast changing battles do have some intrinsic richness to them that synthetic modelling fails to capture, we contend that data induction techniques can be successfully used to generalize combat behaviors. This paper reports the use of neural networks as a computer-based adaptive induction algorithm to understand and uncover ground combat behaviors. Experiments with neural networks using tank movement data from the National Training Center (NTC), Fort Irwin, demonstrate that a two-dimensional cognitive map of closely task organized units can be derived. The findings seem to confirm our behavioral theory that tank commanders (i) are mission-driven, (ii) act as an integral part of their platoon, (iii) perform sequential decision making to determine their next moves, and (iv) when isolated, extemporaneous behaviors may take precedence over normative group behavior. Once trained, a neural-network based model of closely task organized units can be used to predict the itinerary sequences of a tank given its initial geographic position. The findings of this study are being used to support the route determination process within the Single Exercise Analysis Station (SEAS) prototype of the Enhanced Combat Training Center Analysis and Training Methodology (ECATM) research. The goal of the ECATM project is to improve the performance of scenario generation for Janus(A).

Keywords: Combat modeling, Knowledge exploration, Inductive reasoning, Applied Artificial Intelligence, Neural Network, Cognitive Mapping

1. Introduction

In a high-tech ground battle or battle exercise, it is expected that concepts of operations, weapons technology and fighting capabilities of both forces, terrain and weather conditions, individual and collective mental attitudes of engaging troops *do* influence troops' behavior - although with a varying degree of intensity. To emulate this complex reality, combat simulators such as the US Army's Janus(A) combat model are equipped with algorithms to represent warriors' behavior in *typical* combats. Assumably, these algorithms rely on parameters that symbolically represent universal constants of human behaviors, e.g., the proven fighting doctrines and techniques. Furthermore, to cover the various battle contexts that might arise, combat simulators also provide calibration mechanisms for adjusting simulation parameters (for example, see Zyda and Pratt, 1991; Culpepper, 1992, Branley, 1992). For such a calibration to be effective, it has to be performed by well-trained and experienced analysts. Such an exercise *analysis* is time-consuming, subject to human errors, and runs the risk of being incomplete (Tversky and Kahneman, 1974), thus reducing the prediction power of combat simulators.

To circumvent this problem, this paper seeks a *behavioral* rather than analytical representation of the tanks in a battle field. Particularly, and as an effort towards using machine learning techniques for analyzing actual combat behaviors, we propose a neural network (NN) algorithm to capture the actual selection of routes by tank commanders when confronted by perpetually novel and evolving combat situations. When the quality of the data permits, the proposed computer-based adaptive algorithm can next be used to predict the itinerary sequences of a tank given its first position.

The paper is organized as follows. Section 2 introduces a dynamic perspective of the route determination process. Sections 3 and 4 present neural networks as an alternate approach to model closely tasked organizations. Section 5 describes neural network methodology and experimental procedures. It then proposes a cognitive map as

an internal representation of tank commanders' behaviors. Summary of findings and recommendations for future research are provided in Section 6.

2. A Behavioral Model of Closely Task Organized Units

All combat engagement should be the result of a well-defined mission (i.e., tactical goal expressed by mission statement) and a well-thought action plan (i.e., tactical actions). According to the U.S. Army doctrine (FM17-15, 1987), a tank commander should determine his route according to the following major principles:

1. Follow the route determined by the concept of operation;
2. Employ unit movement techniques and drills to balance speed with likelihood of enemy contact;
3. Use the terrain and natural or man-made cover and concealment to mask his weapon system from enemy observation.

It is expected that trained troops – while engaging in combat – should adhere as closely as possible to the concepts of engagement laid out by high-level command. However, actual combat behaviors might deviate from the planned ones, including significant departures from company commander's intent and execution plan. For example, tank commanders are trained that "what can be seen can be killed," so the use of cover and concealment is key to survival on the modern battlefield. When a tank is required to cross open areas, speed and overwatch techniques are used. Factors governing a tank commander's movement include his vehicle's position, route, enemy positions, and his vulnerability. It can be observed that in actual combat situations, tank commanders exhibit the following behaviors when they choose a route:

1. Tank commanders are mission- or goal-oriented. They seek to move as fast as their mission and battle conditions allow to their assigned destination. Drills are used to minimize command and control problems inherent in battle.

2. Tank commanders act as an integral part of their platoon. They are trained on movement techniques and drills which balance speed, use of terrain, and likelihood of enemy contact. They maintain visual contact with other tanks that belong to their platoon. As battle progresses, they adjust their position relative to those of the platoon.

3. Decisions pertaining to route adjustments are sequential. There is an implicit behavior to reject inconsistent moves that do not support the mission.

4. When all communications are lost, extemporaneous behaviors from isolated individual tankers may take precedence over normative group behavior.

As discussed earlier, we contend that route determination, more often than not, is a dynamic and real-time reasoning process with incomplete and quite possibly inexact information. As the battle unfolds, each time slice can be perceived by the engaging tank commander as a life-threatening crisis that forces him to re-evaluate his next movement. The quality of the sequential and dynamic route determination process depends on a large number of factors - particularly, his ability to make use of his knowledge and experience to quickly assess battle situations.

3. Route Determination Paradigms

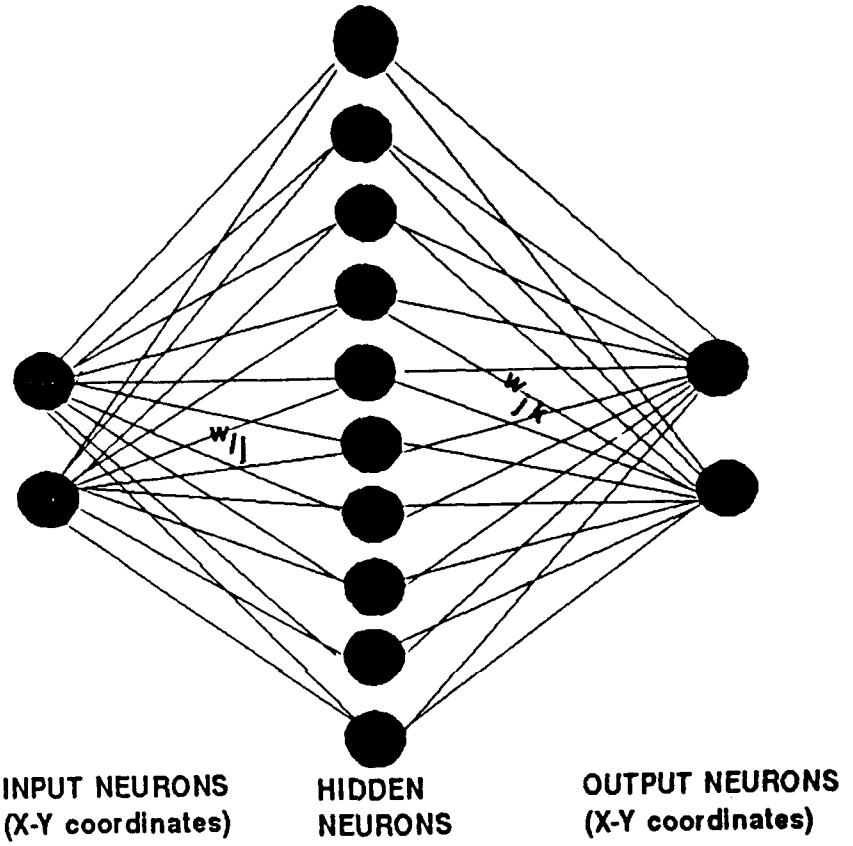
3.1 Deductive and Inductive Approaches to Route Determination

As a decision problem, a closely tasked organization can be determined by using one of the following two paradigms. The deductive approach tries to explain phenomena in terms of causes and effects. All relevant factors that could lead to the construction of

a route should be taken into consideration. Once all hypotheses are formulated and required data gathered, models will be used to predict tank movements. Conversely, the inductive approach conjectures that, in some complex situations such as the route determination process, it would be impossible to model all direct causal relationships due to incomplete, uncertain and dynamic information. To circumvent the difficulty in applying analytical reasoning using quantitative algorithms, the inductive approach hypothesizes that there is a lot to learn from those tanks that successfully make it through to their planned destination. It is believed that "lessons" can be learned by acquiring, processing and refining "knowledge" from actual routes of the mission-accomplished tanks. Hunt (1982) observes that humans possess a "natural" form of reasoning that works surprisingly well in uncertainty. Natural reasoning exploits experience and analogy to reach plausible conclusions. Patterns of a problem are analyzed and compared to previous experiences in an attempt to search for similar circumstances and comparable solutions - in form of "educated guesses". Advocates of this biological approach recognize a strong connection between the structure of the human brain and the ability to reason. The remaining part of this paper describes the use of an artificial neural network (NN) as an analog of the human brain.

3.2 A Brief Description of Neural Networks

A neural network is a system consisting of a number of simple, highly interconnected homogeneous processing units called neurons (see Figure 1). Each neuron is a simple computational device that continuously reacts to external inputs. This reactive behavior can be modeled by relatively simple mathematical functions (For a survey of mathematical functions for neural nets, see for example Hecht-Nielsen, 1988). Typically, a neuron receives input signals from other neurons, aggregates these signals based on an input function, and generates an output signal based on an output or transfer function. The interconnections between neurons is represented by a weighted directed graph, with nodes representing neurons, and links representing connections. The relative



Legend: w_{ij}, w_{ik} : connection weights

Figure 1. A Neural Network Architecture for Tank Routes

importance of the link between two neurons is measured by the weight assigned to that link. A crucial problem in training neural network is to determine a set of weights assigned to the connections that best map all input units to their corresponding output units. In other words, the learning process can be seen as a non-linear optimization problem that minimizes output differences. There are a number of algorithms that can be used to minimize output differences. The back propagation technique is presently the most popular one. Iteratively, it assigns weights to connections, computes the errors between outputs and real data, propagates these error information back, layer by layer, from the output units to the input units, and adjusts the weights until errors are minimized. The back propagation mechanism does not guarantee an optimal solution. However, various experiments reported by Rumelhart et al. (1986) and by other researchers (Maren et al., 1990; Freeman, 1991) suggest that the algorithm provides solutions that are close to the optimal ones.

4. A Neural Network Based Adaptive System for Route Determination

The purpose of this experiment is to develop and calibrate a learning algorithm for a platoon faced with tank movements with initially unknown and random consequences. We assume that the tank commander is able to maintain a high level of situational awareness to continuously adjust his route. However, he is facing a problem of iterated choice under varying degrees of uncertainty (i.e., fog of war). He chooses one of many feasible routes on a "trial" basis, observes the consequence(s) or benefit(s) of that move, and continuously adjusts the tank's direction and speed.

We propose a complex adaptive system for route determination for a tank based on the analysis of its platoon's behavior. The system is complex in that its behavior (i) is based on the dynamic movements of individual tanks that belong to a formation; (ii) exhibits many levels of aggregation and interaction, and (iii) is derived from actual route

data without a detailed knowledge of how each route had been chosen. The system is also an homuncular one in that it learns only from the past experiences of its own platoon, and environmental factors - such as concepts of engagement, terrain and weather conditions, enemy powers, etc. - are somehow embedded in past performances.

Such an adaptive system usually operates far from a global optimum. However, actual data do have some intrinsic richness to them that synthetic modelling could not replicate. Also, it would be easier to capture behaviors that inherently embrace analytical reasoning than to synthetically model the reality that includes, among numerous other factors, human behaviors.

Furthermore, we believe that by observing victorious tanks that successfully made it to destination, sample patterns could be molded and memorized for later use. Learning from these patterns should provide faster and more sensible cues.

Figure 1 describes a simple structure of a neural network designed to learn and simulate the behavior of tank commanders of a platoon in combat. The network is defined by (i) the interconnection architecture between the processing elements - i.e., timely positions of different tanks of a platoon (ii) a transfer function that determines the processing rules, and (iii) learning laws that dictate changes in the relative importance of individual interconnections. Once the system is successfully trained such that it is able to represent the structure and dynamics of actual tank movements, it can be used to simulate/predict the route of a tank given its original geographic position.

5. An Experiment with the Neural Network Model for Route Determination

5.1 Data and Procedures

For the purpose of this experiment, the actual routes of eight tanks in a battle exercise conducted at the National Training Center, Ft. Irwin, were used to train the network model. The tanks were part of a company whose mission was to reach their

destination located approximately 9 kilometers North-East of their initial positions. The platoons moved towards their goal expecting possible contact with the opposing force. All eight tanks achieved the goal.

To emulate the tank commanders' behavior, a neural net was constructed. It has 2 input nodes representing the geographic (latitude/longitude) coordinates of each of the 8 tanks; 2 output nodes representing the geographic coordinates of the subsequent position of a typically behaved platoon tank; and 10 hidden neurons impersonating the internal representation of the perceived environment by the tank commanders. Figure 2 plots the routes of the eight tanks. Each route is represented by forty-three coordinates taken at five minute intervals, beginning with the point of departure and finishing with the destination point.

The back propagation technique was used as the input/output transfer function to determine the relative importance of the interconnections between tank coordinates over time. The network was successful trained to 94% of the training facts after 202 passes, with a training tolerance of 0.1. As expected, the network could not be trained with no tolerance (i.e., training tolerance = 0), because of the noise (stochastic or other) depicted in the routes; i.e., in some portions of the routes, tanks seemed to move slightly to directions other than the intended one toward the planned destination. Figure 3 shows routes simulated by the trained network. The simulated routes retrace with a high level of accuracy the actual routes.

5.2 Testing of Tank Commanders' Behaviors

The successfully trained network could be used to simulate different tank movements given the original position. In this section, we attempt to relate the simulation results of our neural network to the combat behaviors of tank commanders presented in Section 2.

1. *Tank commanders are mission-oriented.* Simulated platoon routes do result in a standard asymptotic pattern converging towards the final destination.

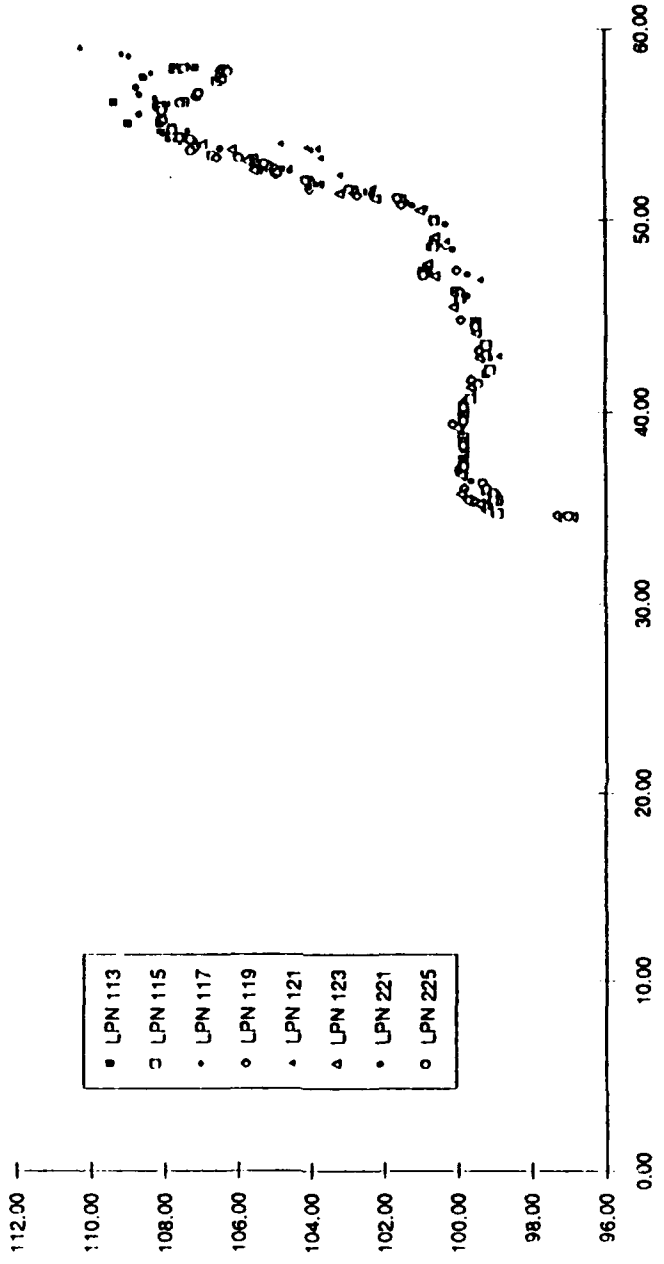


Figure 2. Actual Tank Routes

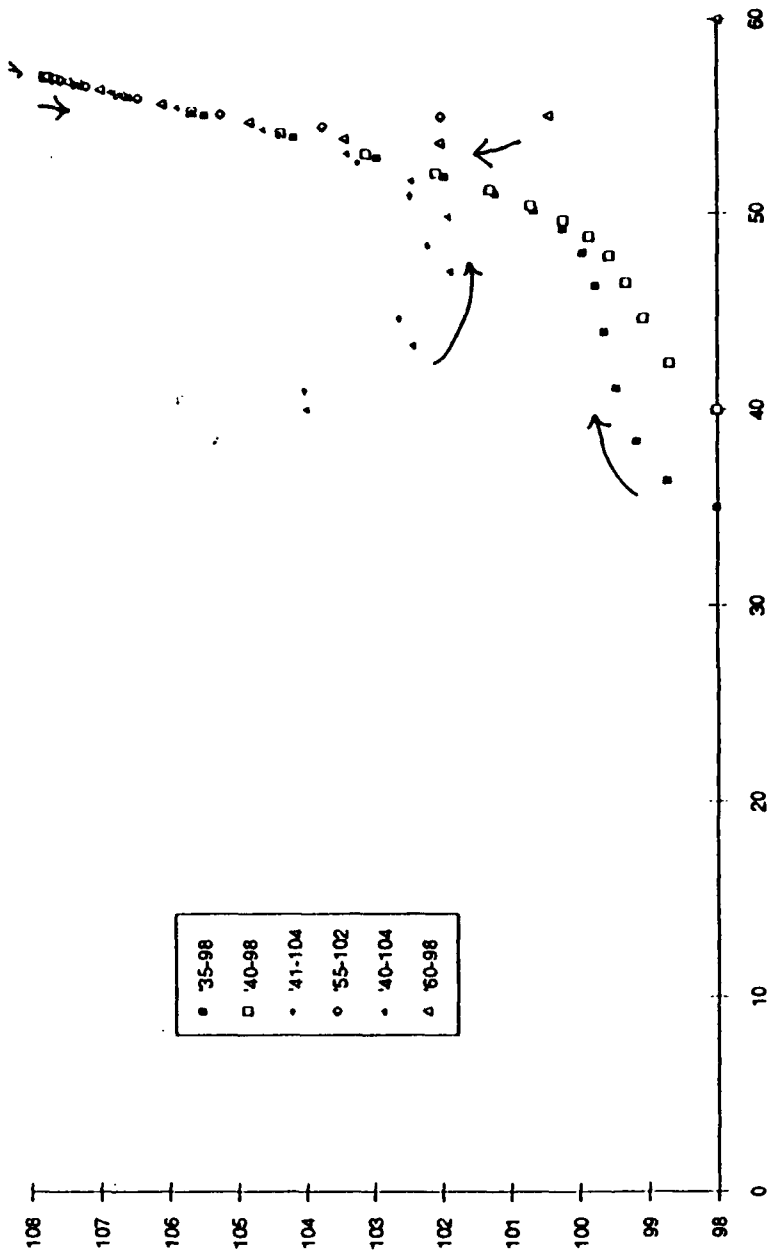


Figure 3. Simulated Routes using a Trained Neural Net

As illustrated in Figure 3, 7 simulated tanks were positioned at various starting points. The routes suggested by the trained network mimicked well the actual counterparts. For example, tanks 35-98 (i.e., with the initial position on the map at 35km on the East-West axis, and 98km on the North-South axis) and 40-98 started at the same positions of actual tanks. The simulation provided the routes *and* the relative speeds to reach destination. Tanks were expected to move quickly at the beginning of their mission, and then to slow down as enemy contact could occur in the central valley, and to gradually reach their target.

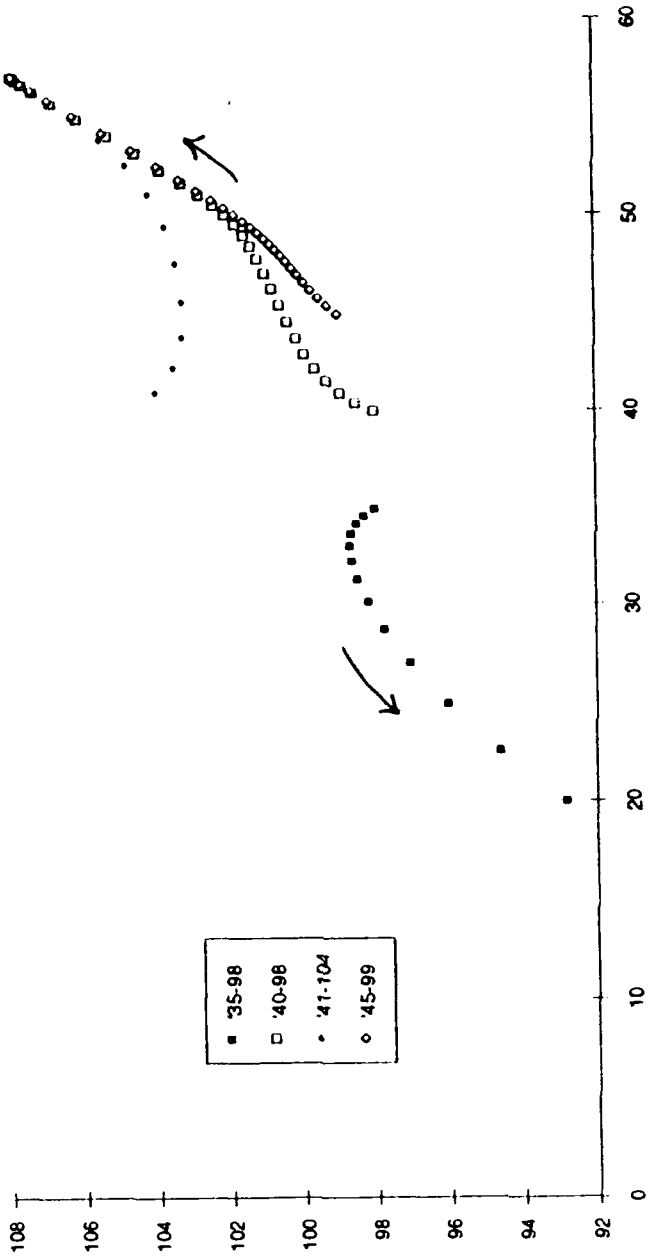
Of particular interest, we simulated the tanks positioned at locations different than the ones used to train the network, especially the one situated at the northern region of the destination zone (55-112). The trained network suggests a short route that leads directly to the intended goal. To further test the mission-driven behavior, we purposely initiated two tanks (40-104; 41-104) from a "no-go" terrain - even though we knew that, in practice, there would be no tanks at that hilly position. The trained network managed to guide them towards the goal area. More importantly, it seemed to recognized the terrain condition. Instead of going straight to the identified goal, the suggested route would take the tank quickly out of the "suspected obstacle" and guide it through the safe contour. Eventually, all tanks stopped once they reached the goal. This hints the stability of the goal state.

2. *Tank commanders consider themselves as an integral part of their platoon.* To test the effect of team coordination, we picked a tank positioned approximately 3 miles south of the rear end of the platoon positions. The tank apparently acted as it recognized that it did not belong to the platoon. It headed for another direction. Similar positions were tested and similar results were obtained. Psychologists who used the Hopfield network (an earlier neural network technique to mimic associative recall) have discovered the same phenomenon (See for example, Hecht-Nielson, 1989). They would argue that the neural net recognized that the tank was not in the vicinity of other tanks - a situation it had never

observed before -, thus decided to generate another goal for that "exotic" tank. Suggesting alternate goals is a innovative approach that learning systems could provide.

3. *Tank commanders perform sequential decision making.* The trained neural net simulated the route one five-minute step at a time. After each move, each re-trained itself, learned from its past, and decided on the next move. The trained net is feed forward. The output is then fed back as the next input, but no retraining occurs. (As can be seen in Figure 3, the trained neural net can "look farther ahead"; a similar phenomenon was discovered by Hutton and Sigillito, 1991). Seemingly, once the system discovered the goal, it tried to accomplish its mission while minimizing its cognitive effort. Wherever possible, fewer and faster steps were identified to reach the final target. Eventually, all of them conversed at the intended target.

4. *Tank commanders reject inconsistent movements.* In testing the trained network, we had no problem re-tracing the tanks that were stationed at their intended starting positions. The simulated routes were determined as expected. We were not sure, however, how to position the tanks to start at "unconventional" starting locations and interpret the respective routes suggested by the trained neural net. In particular, we wanted to train the neural net to recognize "no-go" terrain so that it can "penalize" all attempts to start a tank at an infeasible position. Note that this is a theoretical issue for, in reality, no analyst or company commander would assign tanks at impossible location. A set of feasible routes were artificially created to emulate the zone of "go" terrain and added to the original data set (Figure 4). As might be expected, the system was trained with a much lesser degree of confidence. In the simulation after training, the tanks seemed to wander around with much less determination than in the net trained with only real data.



Legend: Arrows indicate direction of simulated routes

Figure 4. Trained Network with Feasible Routes

5.3 A Two-Dimensional Cognitive Map of Closely Tasked Organizations

Cognition is the process by which external or sensory inputs are perceived, processed, stored, recovered, and used (e.g., Neisser, 1967). In this section, we attempt to synthesize the findings of our experiment by constructing a cognitive map used by tank commanders during combat engagement. The cognitive map provides a symbolic representation of how tank commanders see the battle (see 5.2), yet is capable of using this internal representation to solve dynamic problems at hand. The internal representation is a mixture of knowledge (i.e., that which is known to be true about something), and beliefs (i.e., that which is believed to be true) – a common phenomenon discovered in cognitive science (Konolidge, 1986; Hintikka, 1962). Figure 5 is a cognitive map of the tanks used in this experiment, presented on a two-dimensional geographical space.

The neural network learned that there was a *goal/mission* – better yet, a stable and purposeful one. Tanks that are seemingly not part of the mission should look for other goals (i.e., *alternate goals* in the map). Environmental factors are embedded in the way tanks moved. Tank movements and speed reflected the terrain and weather conditions, as well as enemy forces. Emulation of "known" routes displays a *habit formation* characterized by an exacting behavior with repeated exposure. Emulation of "unknown" routes suggests that new behavior can be *"learned"* to face with new contexts. The analysis of these confirmed and revealed behaviors could help discover *unapparent knowledge*. As an example, the suggested routes to reach the destination systematically showed a consistent detour suggesting a forbidden or no-go zone on the left-hand side.

5.4 Discussion

The experiment conducted in this paper suggests that a simple dynamic system could represent a complex reality such as tank commanders' behaviors in battlefield. For a *short-term, well-focused* decision problem such as the process of route determination, the data induction technique helps understand behaviors without requiring full

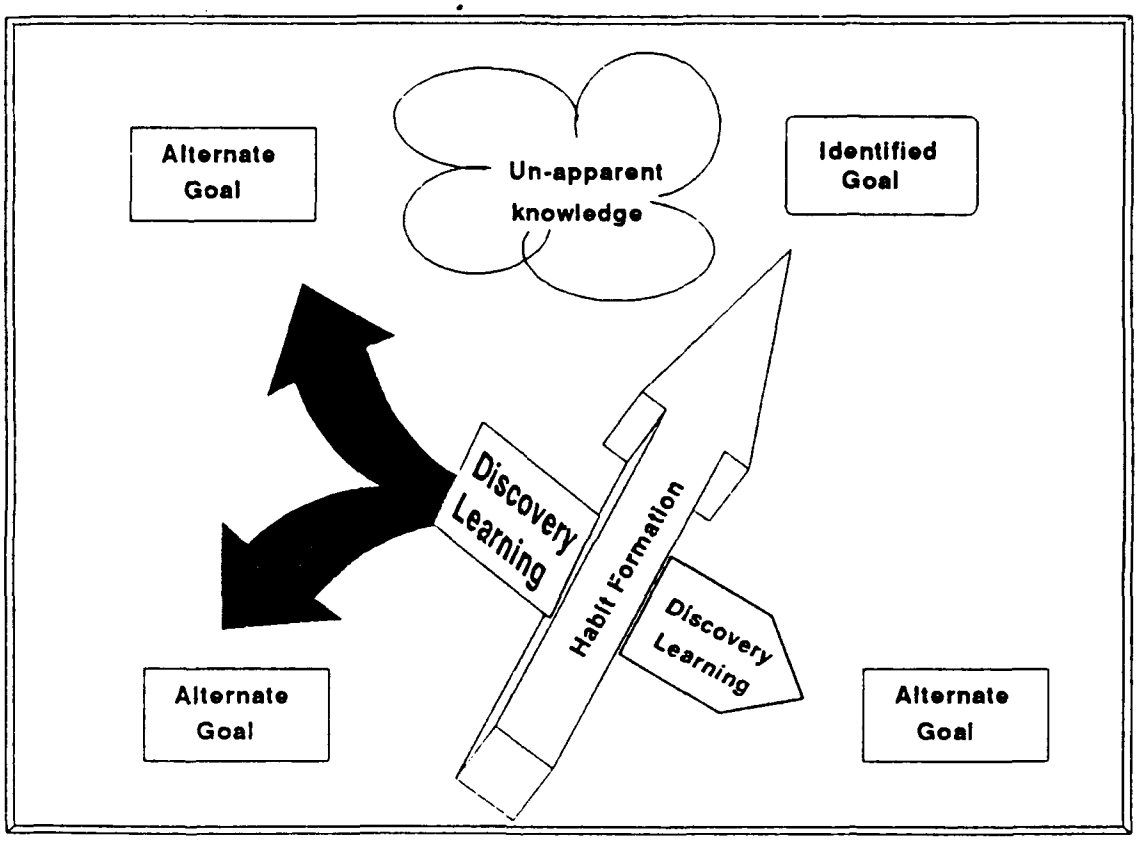


Figure 5. A Neural-Network Based Cognitive Map of Tank Commanders

understanding of all behavioral elements involved and their relationships, and the statistical techniques necessary to analyze these elements. The focus is on analyzing the *pattern(s)* of the actual data.

The approach taken here is certainly not new, for it is well-known in the fields of artificial intelligence and psychology. However, the cognitive map derived from neural network technique proposes an innovative way of analyzing, interpreting tank commanders' behaviors, and determining sensible tank routes.

From a practical viewpoint, the derivation of a cognitive map using neural network approach presented in this paper provides a number of benefits. First, NN techniques use algorithms capable of solving non-linear problems; a typical issue in route determination. Second, the level of complexity required to determine route with NN is significantly less than that required by conventional statistical/operations research approaches. Third, the NN technique is more cost-effective (See for example, AIRMICS report, June 1991).

It is widely acknowledged among AI researchers that neurocomputing attempts only to provide quick and rapid solutions, and combining the processing and learning capabilities with conventional analytical tools would provide complementary reasoning mechanisms to tackle perplexing problems. The proposed technique can be combined with conventional techniques such as simulation models and rule-based systems to provide enhanced analytical capabilities for determining troops' movements. Although further knowledge exploration and testing are required, the findings in this experiment suggest that the NN model seems to produce results that are at least as robust as those obtained from conventional techniques. As such, the cognitive map proposed in this paper is a concrete step towards this effort.

6. Summary and Recommendations for Future Research

The purpose of this paper was to suggest the use of neural network algorithms as a data induction technique to reproduce battlefield troop behaviors. The experiment reported suggests that neural networks as artificial learning agents could be trained from actual tank commanders' behaviors. This paper also assumes that troop behaviors are homogeneous in that tank commanders would follow the steps of their victorious peers. As such, the proposed model does not address issues related to the group/coordination behaviors of tanks within a platoon. The observed behaviors do reflect movement techniques specified in FM17-15, as well as those that deviated from the doctrinal level to account for battle situations. Not only does the learning behavior of the neural net reproduce fairly well circumstance-dependent platoon behaviors, it also reveals the possible existence of inconsistent or random behaviors of platoons.

We could claim that the major benefit of such an approach is to furnish predictions based on actual rather than pure doctrinal behavior; we assume that victorious tank commanders successfully combined doctrinal strategy with contextual tactics. As such, the neural net provides a convenient dynamic representation that can be inserted into theoretical models. Discovery techniques do not make any assumptions regarding the functional relationships contained within the data. The proposed neural network model could be used as a benchmark for, and provide insights into, existing NTC data and Janus(A) scenarios.

The findings reported in this report should be at best considered preliminary results that call for a more extensive testing of the proposed neural network with various troop formations in a battle exercise. Tests using more data inputs and various networks archictectures are being conducted to enhance the accuracy of the simulation. If a neural model can help reconstruct the behaviors of tanks commanders, it could be used to verify the theory embedded in combat simulators. Furthermore, it could be used to guide unmanned autonomous tanks in combat.

REFERENCES

AIRMICS, *Application of Neural Networks for the Extraction and Characterization of Knowledge Contained in Databases*, ASQB-GM-91-013, AIRMICS, Georgia Institute of Technology, Atlanta, GA, June 91

Branley Jr., W.C., *Modeling Observation in Intelligent Agents: Knowledge and Belief*, Master Thesis, Naval Postgraduate School, Monterey, CA, 1992.

Culpepper, M., *Tactical Decision Making in Intelligent Agents: Developing Autonomous Forces in NPSNET*, Master Thesis, Naval Postgraduate School, Monterey, CA, 1992.

Freeman, James and David Skapura, *Neural Networks, Algorithms, Applications, and Programming Techniques*, Addison-Wesley, Reading, Mass., 1991.

FM17-15, *Tank Platoon*, Cdr, USAARMC, ATTN: ATZK-DS, Fort Knox, KY 40121-5000.

Hecht-Nielsen, R., "Theory of the Back Propagation Neural Network", *Neural Network*, 1,131, 1988.

Hintikka, J., *Knowledge and Belief*, Cornell University Press, 1962.

Holland, J.H. et al., *Induction: Processes of Inference, Learning, and Discovery*, Cambridge, MIT Press, 1986.

Holland, J.H. and J.H. Miller, "Artificial Agents in Economic Theory", *American Economic Association Papers and Proceedings*, Vol 81, NO. 2, pp. 365-370, May 1991.

Hunt, M., *The Universe Within: A New Science Explores the Human Mind*, Simon and Schuster, New York, 1982.

Hutton, Larrie and Vincent Sigillito, "Experiments on Constructing a Cognitive Map: A Neural Network Model of a Robot that Daydreams", *Proceedings of the 1991 IEEE Conference on Artificial Intelligence*, pp. 223-228.

Konolidge, K., *A Deduction Model of Belief*, Pitman, London, 1986.

Maren, Alianna, Craig Harston, and Robert Pap (Ed.), *Handbook of Neural Computing Applications*, Academic Press, San Diego, 1990.

Neisser, U., *Cognitive Psychology*, New York, Meredith Publishing, 1967.

Turing, Alan M., "Can Machine Think?" in John R. Newman, ed. *The World of Mathematics*, Vol. 4, New York, Simon and Schuster, 1956, 2009-2123.

Tversky, Amos and Daniel Kahneman, "Judgment under Uncertainty: Heuristics and Biases", *Science*, 1974, pp. 1124-1131.

Zyda, M.J., and D.R. Pratt, "NPSNET: A 3D Visual Simulator for Virtual Exploration and experimentation", Digest of Technical Papers XXII, *SID International Symposium*, May 1991.

SUMMARY OF DISCUSSION GROUPS

Edmund Thomas
Jan Dickieson
Navy Personnel Research and Development Center

Group members included representatives from all services, both researchers and managers. The manpower and personnel discussion group noted the following:

As the Clinton Administration begins to place increasing emphasis on dual-technology applications, military researchers must be more sensitive to potential commercial applications for the products of their efforts--

- Neural-network delivery (implementation) hardware/software
    - Components and full systems for general/specialized applications (motion detection)
    - Linkages to other sensory systems, e.g. optical sensors
    - Embedded machine controllers
- Network development tools
    - Design and debug of applications networks
    - Generate and test complex networks (network compilers and assemblers)
    - Evaluate hardware such as advanced graphics systems for network displays
- Network applications
    - Development involving back-propagation, Hopfield networks
    - Detection of hidden trends in banking financial data, insurance, investments
    - Applications to education and social problems

There are possibilities to be explored where we can model intelligence along the following lines:

- An event occurs in the environment
- The system detects an external event (transduction) which causes internal change to the system
- Internal event causes selection and execution of a system response
- Execution of response(s) causes events (changes) in external environment
- The cycle repeats in an iterative manner

The store of knowledge may be acquired as a 'hard wiring' form, such as the fixed structure of knowledge bases used in today's artificial intelligence expert systems or by learning, in which a system acquires knowledge and is capable of making decisions through interaction with its environment.

One prime difficulty for the military laboratories is 'inertia' in the form of reliance by sponsors on what is known now (the status quo) and the reluctance to invest in new technologies. Downsizing of the military may have a significant impact on the funds likely to be invested in new technologies.

The training discussion group noted the following:

Current applications in training include:

- scenario generation
- intelligent tutoring

There are a number of research needs:

- more rigorously controlled testing for modeling of human behavior
- development of the capability to better explain network results
- improvements in the preprocessing of data

There are many areas where applications of neural networks and related technologies might prove beneficial. These areas include:

- small targets that would react to simulation input as humans would react
- measuring expertise
- intelligent tutors
- building model trainees

There was consensus that the conference fulfilled its original objectives. It was felt that participants are involved in a field where knowledge gains occur rapidly; therefore, the value and importance of holding an annual meeting which emphasizes behavioral sciences is tremendous.

Conference Staff

Commanding Officer, NPRDC

CAPT John McAfee

Chairperson

Mr. Edmund Thomas

Coordinator

Dr. Jules Borack

Operations Committee

CDR. Michael Brattland
Mr. Edmund Thomas
Ms. Norma Zaske
PNC Dusty Porter
STSCS Jack Banks
Ms. Betty Griswald
Ms. Arneva Johnson
Ms. Rosa Broadway

Finance

Mr. Charles Bigsby

Security

Mr. Richard Plumlee

Publications/Graphics

Ms. Marci Barrineau

Hospitality

Mr. Edmund Thomas

Conference Participants

Participant and Organization	Phone(s)
Aleshunas MAJ (USA), John USA Reserve Personnel Center Attn DARP-ZAP-TR 4700 Page Boulevard St Louis, MO 63132-5200	314 538-2340 DSN: 893-2340
Allard, Terry Office of Naval Research, Code 1142 800 North Quincy Street Arlington, VA 22217-5660	703 696-4502 DSN: 226-4502
Bui, Tung Naval Postgraduate School Information Technology, Code AS/BD Monterey, CA 93943	408 646-2630 DSN: 878-2630
Byrne CAPT(USMC), Brian Naval Postgraduate School Code 36 NPGS Monterey, CA 93940	408 656-2536 DSN: 878-2536
Callahan, Janice D. Callahan Associates 874 Candlelight Place La Jolla, CA 92037	619 488-0130
Dickieson, Jan NAVPERSRANDCEN, Code 132 53335 Ryne Road San Diego, CA 92152-7250	619 553-9270 DSN: 553-9270
Duffy, Lorraine NCCOSC/NRAD, Code 442 271 Catalina Boulevard San Diego, CA 92152-5000	619 553-9222 DSN: 553-9222
Fleming, Jimmy L. Armstrong Laboratory Code AL/HRTI Brooks AFB, TX 78235-5601	512 536-2034 DSN: 240-2034

Participant and Organization	Phone(s)
Folchi, John NAVPERSRANDCEN, Code 122 53335 Ryne Road San Diego, CA 92152-7250	619 553-7750 DSN: 553-7750
Goldberg, Lawrence Economic Research Laboratory 11429 Purple Beech Drive Reston, VA 22091	703 758-1431
Greenston, Peter Army Research Institute, Code PERI-RG 5001 Eisenhower Avenue Alexandria, VA 22333	703 274-5610 DSN: 284-5610
Grobman LT(USAF), Jeff Armstrong Laboratory Code AL/HRMM Brooks AFB, TX 78235-5352	210 536-3551 DSN: 240-3551
Hawkins, Harold Office of Naval Research, Code 1142 800 North Quincy Street Arlington, VA 22217-5660	703 696-4323 DSN: 226-4323
Hurwitz, Joshua B. Armstrong Laboratory Code AL/HRMIL Brooks AFB, TX 78235-5601	210 536-3713 DSN: 240-3713
Lewis, Greg W. NAVPERSRANDCEN, Code 134 53335 Ryne Road San Diego, CA 92152-7250	619 553-7709 DSN: 553-7709
Looper, Larry T. Armstrong Laboratory Code AL/HRMM Brooks AFB, TX 78235-5352	210 536-3648 DSN: 240-3648
Makeig, Scott Naval Health Research Center PO Box 85122 San Diego, CA 92186-5122	619 553-8416 DSN: 553-8416

Participant and Organization	Phone(s)
Marshall, Sandra P. San Diego State University Department of Psychology San Diego, CA 92182	619 594-2695
Mcbride, James Human Resources Research Organization 6430 Elmhurst Drive San Diego, CA 92120	619 582-0200
Pytel LT (USN), Dennis Naval Postgraduate School Code 36 NPGS Monterey, CA 93940	408 656-2536 DSN: 878-2536
Ramah, Gary John Armstrong Laboratory Code AL/HRTI Brooks AFB, TX 78235-5601	210 536-2034 DSN: 240-2034
Russell LT (USN), Bradley S. Naval Postgraduate School Code 36 NPGS Monterey, CA 93940	408 656-2536 DSN: 878-2536
Ryan-Jones, David L. NAVPERSRANDCEN, Code 134 53335 Ryne Road San Diego, CA 92152-7250	619 553-7710 DSN: 553-7710
Sands, W. A. (Drew) NAVPERSRANDCEN, Code 12 53335 Ryne Road San Diego, CA 92152-7250	619 553-9266 DSN: 553-9266
Scheines, Richard Carnegie Mellon University Department of Philosophy Pittsburgh, PA 15213	412 268-8571
Schulz, David S. Naval Postgraduate School Code 36 NPGS Monterey, CA 93940	408 656-2536 DSN: 878-2536

Participant and Organization	Phone(s)
Shoemaker, Patrick NCCOSC/NRAD Code 552 271 Catalina Boulevard San Diego, CA 95152-5000	619 553-5385 DSN: 553-5385
Sinaiko, H Wallace Smithsonian Institute 801 North Pitt Street Alexandria, VA 22314	202 357-1829
Smith, David E. NCCOSC/NRAD, Code 444 271 Catalina Boulevard San Diego CA 92152-5000	619 553-5209 DSN: 553-5209
Smith LTC (USA), Gaylon Headquarters Department of Army ODCSPER DAPE-ZXP, Room 720, Pentagon Washington Dc 20310	703 697-6700 DSN: 227-6700
Sorensen, Stephen W. NAVPERSRANDCEN, Code 112 53335 Ryne Road San Diego, CA 92152-7250	619 553-7656 DSN: 553-7656
Sorenson, Richard C. NAVPERSRANDCEN, Code 01 53335 Ryne Road San Diego, CA 92152-7250	619 553-7813 DSN: 553-7813
Su, Yuh-ling NAVPERSRANDCEN, Code 11 53335 Ryne Road San Diego, CA 92152-7250	619 553-0729 DSN: 553-0729
Thomas, Edmund D. NAVPERSRANDCEN, Code 01D 53335 Ryne Road San Diego, CA 92152-7250	619 553-7820 DSN: 553-7820
Trejo, Leonard J. NAVPERSRANDCEN Code 134 53335 Ryne Road San Diego, CA 92152-7250	619 553-7711 DSN: 553-7711

Participant and Organization	Phone(s)
Vickers, Ross R. Naval Health Research Center PO Box 85122 San Diego, CA 92186-5122	619 553-8417 DSN: 553-8417
White, Halbert University of California, San Diego Department of Economics 0508 La Jolla, CA 92093	619 534-3502
Wiggins, Vince RRC Incorporated 3833 Texas Avenue Suite 285 Bryan, TX 77802-4039	409 846-4713
Wilkins, Chuck A. Baylor University Department of Psychology Waco, TX 76798	817 754-4688 817 755-2961
Wolfe, John H. NAVPERSRANDCEN, Code 121 53335 Ryne Road San Diego, CA 92152-7250	619 553-9251 DSN: 553-9251

Author Index